Managing Uncertainty in Airline Ground Operations

Stochastic Staff Scheduling

MSc Industrial Engineering and Management
Jeroen Assink
March 2025

KLM Royal Dutch Airlines

UNIVERSITY OF TWENTE.

Master thesis Industrial Engineering and Management

Managing Uncertainty in Airline Ground Operations: A Stochastic Staff Scheduling Approach

Author:
J.G. Assink (Jeroen)
j.g.assink@student.utwente.nl

This is a public version of the thesis, data is modified, names are changed, and scales are removed from the figures. A confidential version of the thesis can be requested at the author.

Koninklijke Luchtvaart Maatschappij
Amsterdamseweg 55
1182 GP Amstelveen
(020) 649 9123

University of Twente
Drienerlolaan 5
7522 NB Enschede
(053) 489 9111

Supervisors KLM
Dr. R. Gupta (Rohit)
Dr. T. Pippia (Tomas)

Supervisors University of Twente
Dr. D.R.J. Prak (Dennis)
Prof.Dr.Ir. M.R.K. Mes (Martijn)
U.D.D.M. Dahanayaka (Mahekha)

Preface

Dear reader,

In front of you lies my master thesis: "Managing Uncertainty in Airline Ground Operations: A Stochastic Staff Scheduling Approach". This research is performed at The Koninklijke Luchtvaart Maatschappij (KLM) and marks the end of my time as a student Industrial Engineering and Management at the University of Twente.

At KLM, I was welcomed into the Data Science team of ground operations, consisting of multiple (extremely smart) colleagues, willing and able to understand my work and help me out. Whether it was answering a quick question, making an effort to bring me into contact with the right people, or sitting we me for an hour to work out a detail of my thesis, the colleagues were always available for help. A special thanks to my supervisors Tomas Pippia and Rohit Gupta, who helped me get acquainted with KLM, gave me valuable feedback and could think along about possible solutions to the problems that arose. Another special thanks goes to the Data Scientists of the Tactical planning team (Joshe Klaver, Mehran Shakarami, Abhishek Bharadwaj), with whom I enjoyed in-depth discussions about the model and its behaviour. Without you, I would not have been able to yield some of the nice results that came.

Furthermore, I would like to express my gratitude to the supervisors from the University of Twente. First of all, Mahekha Dahanayaka, who performed similar research and thus could provide me with a lot of in-depth feedback. Secondly, I want to thank Martijn Mes for his advice on improving the academic level of my thesis. Lastly, I want to thank Dennis Prak for the many feedback sessions and for brainstorming with me about 'cool graphs and figures'. I especially enjoyed the spontaneous five-minute meetings that always turned into 45-minute discussions. Next to that these sessions often helped me in writing my thesis, the informal and off-topic conversations that came with it made the six months great fun as well.

I cannot end without thanking Roy Koers for teaming up with me in almost all project groups during the masters. Even in my thesis, he was a great support by thinking along whilst creating my model and introducing some very valuable new ideas. But most importantly, thank you for making my student time fun and memorable!

I hope you enjoy reading this thesis and that it may contribute to other research as a source of inspiration.

Jeroen Assink
Enschede, 2025

Management Summary

This report presents a study on optimizing ground staff scheduling at KLM. The research addresses the challenges of staff scheduling in the context of airline ground operations, particularly for luggage handling. KLM's current scheduling approach relies on static models and planner experience, improvements can be made by accounting for real-time dynamics such as flight delays and task duration uncertainties. This leads to schedules that are better aligned with the Day of Operation (DoO). When disruptions occur, there is the need for costly last-minute adjustments, including outsourcing tasks, using staff from other departments, or delaying flights. To create a scheduling algorithm that can ensure resilience against these disruptions, we formulated our research question as:

"How can KLM optimize its ground staff scheduling to incorporate robustness alongside effectiveness and efficiency?"

The literature contains research on staff scheduling. However, as far as we know, no literature can provide a model formulation that solves the problem in a stochastic environment, including the exchange of staff between different skills. Therefore, we conclude that a gap in the literature exists.

To incorporate the stochastic nature of tasks, we solve the scheduling problem by formulating a Stochastic Demand Accumulation Model (SDAM). We approximate the stochastic programme using the Sample Average Approximation (SAA) method to minimize shift costs and understaffing costs and use the method to estimate the number of possible DoO scenarios needed in the optimization model to accurately assess the real-life circumstances, we identified forty scenarios as accurate. To create a model capable of generating optimal solutions in reasonable runtime, we transform the current scheduling approach by KLM. The key methods and models developed are the Demand Accumulation Model (DAM), which shifts from assigning individual tasks to aggregating demand per period, allowing for flexible period lengths. The DAM reduces the model size significantly, making it more computationally efficient. Compared to the current Task Assignment Model (TAM), the DAM reduces the number of variables by 99.7% and the number of constraints by 99.9% by using one integer number to represent total demand in set time periods, rather than representing each task separately. We evaluate this method and find that above 99.5% of tasks can be assigned for all tests. The SDAM extends the DAM by incorporating stochastic demand. The SDAM optimizes schedules across different scenarios, improving robustness against unexpected disruptions. A comparative analysis between the DAM and SDAM demonstrates that incorporating stochasticity leads to cost savings between 7% and 35%. Furthermore, an additional test revealed that even with perfect information about task details (where a fully deterministic scheduling approach would suffice) only 7% of further cost reductions could be achieved. This finding highlights that the SDAM already captures the majority of potential savings, significantly improving scheduling efficiency while leaving only a smaller margin for further optimization. Further cost reduction can be achieved by allowing a exchangeability of tasks between as many skills and departments as possible, providing flexibility and thus allowing saving up to 10.2% of costs. Computational performance of the model can be considered good as the model solves to optimality in 25.3 seconds.

We extend the SDAM to allow the model to delay scheduling the break until demand scenarios have been realized. Currently, this does not yield added benefit to KLM due to the conservative approach taken (reducing understaffing by hiring more employees). This results in the SDAM being able to create a solution that is already optimal for each scenario. We believe that input and parameter settings make a difference here, and the delayed break model is preferred over the SDAM since it can yield better results if the input is changed in the future. We further extend the SDAM to a robust optimization model by incorporating certainty parameters specifically. With these parameters, KLM can choose the percentage of security wanted to satisfy demand. Sensitivity analysis of this model shows that improving demand coverage from 90 to 98% will incur the same costs as improving from 98-100%, highlighting the exponential increase in costs. Since understaffing costs are not visible directly but are a consequence of multiple factors, this parameter is hard to estimate. With the estimated understaffing costs of [redacted] we identify that the SDAM covers around 99.5% of demand, highlighting that a conservative approach is recommended.

Based on these findings, the research recommends that KLM implements the DAM for shift scheduling, as it significantly reduces computation time while maintaining a high task coverage rate. The SDAM or the delayed break model should be used to account for variations in demand, leading to more robust schedules and cost reductions compared to deterministic approaches. Task stochasticity should be incorporated by creating real-world scenarios based on observed fluctuations in task duration rather than just deriving scenarios from delays.

This research contributes to KLM by providing a faster scheduling algorithm, with a higher potential by not narrowing down input sets, and the ability to optimize different scenarios combined, making the model resilient to DoO schedule changes. It also introduces employee exchangeability, allowing for more cost-effective combined optimization between different employee groups. However, there are limitations. First, current stochastic scenario calculations only consider plane delays while neglecting other sources of variation. Additionally, the study is limited to only one employee skill (Team Member) in two departments. Further research should focus on extending the experimental analysis with planning more employee skills using the SDAM, explicitly involving unequal skilled employees, to see how unequal relations affect the scheduling process. Lastly, postponing break scheduling until after demand has materialized can improve the optimal schedule and therefore be an interesting topic for future research.

Contents

Acronyms

ALNS Adaptive Large Neighbourhood Search.

DAM Demand Accumulation Model.

DoO Day of Operation.

EVPI Expected Value of Perfect Information.

KLC KLM Cityhopper.

LNS Large Neighbourhood Search.

MIP Mixed Integer Linear Program.

MPS Manpowered Planning System.

MSD Minimal Shift Design.

SAA Sample Average Approximation.

SDAM Stochastic Demand Accumulation Model.

SO Shift Optimizer.

STO Shift Type Optimizer.

TAM Task Assignment Model.

VSS Value of the Stochastic Solution.

1 Introduction

This chapter gives an introduction to the research that is performed at the Koninklijke Luchtvaart Maatschappij. Section 1.1 covers the background knowledge and the origin of the research topic. Subsequently, Section 1.2 describes the approach of this research.

1.1 Research context

This research addresses a staff scheduling problem at an airport, specifically involving ground staff responsible for performing various tasks at the apron, where planes are stationed. Sections 1.1.1 and 1.1.2 provide further insight into the company and the origins and context of this research. Section 1.1.3 explains the focus of the research.

1.1.1 Company introduction

The Koninklijke Luchtvaart Maatschappij N.V. (KLM), established in 1919, serves as the national airline of the Netherlands. Today, the airline operates flights to around 100 European destinations and 70 intercontinental routes (KLM Royal Dutch Airlines, 2023). In 2004, the airline merged with Air France, forming one of the largest airline groups in the world. KLM's revenue is derived from three core activities: passenger transport, freight transport, and aircraft maintenance.

Platform Ground is a sub-department within KLM's Data and Technology division. Its mission is to enhance the customer experience at the airport by supporting operations and optimizing turnaround processes—the procedures involved in preparing an aircraft from arrival to departure. This is done by continuously improving the processes, resources, and tooling used. KLM manages and operates all ground processes in-house. At its home base and hub, Schiphol Airport in Amsterdam, KLM also provides ground handling services to other airlines, making KLM's logistics notably complex.

1.1.2 Motivation for research

In September 2024, KLM management announced a reorganization as a consequence of high costs and shortages in staff (NOS Nieuws, 2024). According to the current information, KLM maintains an employee base of approximately 35,000 (31,000 FTE). Around 15,000 of these employees perform ground operations, making this area crucial for the company. Since many tasks require manual labor, staff scheduling plays a vital role in KLM's operations, making efficient and effective scheduling a challenge.

From a business perspective, labor costs represent a significant direct expense for companies (Akkermans et al., 2021). The aviation industry is characterized by high capital costs and low operating margins, driving airlines to constantly seek time and cost reductions to remain competitive (Cavada et al., 2023). Disruptions of passengers' connections, crews, and ground operations are major sources of cost for airlines (Cook and Tanner, 2015). More specifically, disruptions of ground operations, such as early/late flight arrival or task duration deviations, contribute to delay propagation: the subsequent tasks of the assigned employees are also delayed (Poullet and Parmentier, 2020). That airlines indeed cope with those delays is studied and reinforced by Santos and Robin (2010), during the last decades, flight delays have increased to over 20%. At KLM, this is even a bit worse, in 2023, the airline managed a 77.5% on-time performance (KLM Royal Dutch Airlines, 2023). While KLM, like many airlines, used to focus primarily on balancing revenues and costs, employee availability has now become an additional critical constraint. Staffing challenges (particularly after the COVID-19 recovery) have made employee availability a major issue across the industry (Sun et al., 2023; Scheelhaase et al., 2023). Society for Human Resource Management (2016) shows that 67% of organizations experience moderate to significant productivity impacts due to staff shortages, with 51% of employees reporting increased stress as a result.

KLM experiences the problems explained above and could benefit both financially and in employee satisfaction by constructing a resilient staff schedule. Currently, KLM's planners schedule their staff based on their experience, a set of business rules, and static models, which means that the explained dynamics (flight delays and task duration uncertainty) present at the airport are not considered. As a consequence, the shift schedule does not fit the Day of Operation (DoO) and real-time changes have to be made. Either tasks need to be outsourced, staff from other departments need to assist or tasks (and thus flights) are delayed. The root of this issue lies in the scheduling algorithm, which is not resilient against possible disruptions. While it accounts for efficiency (avoiding overstaffing) and effectiveness (avoiding understaffing), it neglects robustness—its ability to handle uncertainties.

In the literature, this problem is known as the shift scheduling problem, but practical solutions to properly schedule considering robustness are lacking. van den Bergh et al. (2013) extensively reviewed personnel scheduling approaches, examining 306 papers. While some papers discuss staff scheduling under the stated uncertainties and papers discuss staff scheduling in aviation, only $<1\%$ address stochastic environments in aviation, none of which focus on ground staff for passenger flights. This highlights the need for further research in the airport setting. Nowadays, some literature like Hur et al. (2019b) proposes a solution to address the shift scheduling problem under stochastic demand but lacks an implementable solution due to several reasons (which are elaborated upon in Chapter 3).

This research aims to enhance the shift scheduling of the ground staff by anticipating fluctuations and disruptions that may occur in its execution. The expectation is that improved schedules limit the changes that are to be made on the DoO, which improves employee satisfaction and reduces flight delays and costs.

1.1.3 Scoping

Each day, around 620 flights arrive or depart from Schiphol Airport (KLM Royal Dutch Airlines, 2023), requiring coordination of passenger services, maintenance, catering, cleaning, fueling, and (transfer) luggage services for every flight. While all personnel scheduling is essential, this research focuses on luggage handlers as a use case to test potential improvements in scheduling. Luggage handling is a significant operation, requiring around 3,700 fte per week, making it a crucial area for optimization. By focusing on this large-scale workforce, we can evaluate the impact of the proposed scheduling model formulation. Subsequently, the model formulation can be transformed to apply to other departments. We additionally scope on stochastic factors to only include plane delays (explained in Section 2.2).

1.2 Research design

To solve the problem described, we have designed a research plan, detailed in this section. The main research question results from the described problem statement and is formulated as:

"How can KLM optimize its ground staff scheduling to incorporate robustness alongside effectiveness and efficiency?"

A series of sub-questions has been formulated to address the main research question. The initial set of sub-questions focuses on the current state of operations at KLM, aiming to identify the specific processes that require scheduling and investigate how these schedules are currently generated. To answer these, an in-depth investigation is conducted at KLM's homebase, Schiphol Airport. First, we examine which processes are being scheduled and identify the key disruptions affecting the DoO. Following this, we analyze the existing scheduling process to understand the steps KLM follows to create schedules, the requirements of the current scheduling model, and any gaps that need to be addressed. Finally, we determine the most effective points within the scheduling process to introduce robustness.

To answer the main research question, we perform an observation of current processes and an investigation of the tools currently used. Subsequently, insights from stakeholders and available data considering scheduling are used. In Chapter 2 we answer the sub-questions by identifying the activities that are to be scheduled for the DoO, and the methods used for scheduling these activities.

Chapter 2. Which processes are part of the ground operations and what is done to create schedules for these processes?

Ground operations
- What are the current luggage handling processes?
- Which uncertainties are present?

Planning and scheduling processes
- How are the staff schedules currently created?
- What are the current objectives and restrictions of the production schedules?

After analyzing the current situation, we explore the scientific literature to examine the scheduling problem from a theoretical perspective. First, we position our problem in the literature. Then we identify existing research that aligns with the challenges faced by KLM, ensuring that our approach builds upon proven methods. Subsequently, we look into the characteristics and preconditions of the found solution methods. This is done to see whether we can translate the solution to our research. Chapter 3 covers the found literature. The literature displays the existing methods that are used at airports and look into methods from other fields that have the potential to be translated into our problem.

> Chapter 3. What methods for robust staff scheduling are discussed in the literature?
>
> ---
>
> - How can we translate the studied case to a theoretical problem?
> - What is the current state-of-the-art regarding staff scheduling?
> - How can stochasticity be addressed when scheduling staff?

Continuing the research, insights from the literature are applied to the current problem to construct an improved shift scheduling method. First, understanding how to best model uncertainty is critical for enhancing the reliability of schedules in the face of disruptions. Second, adjusting the objective function to incorporate robustness ensures that the new model not only optimizes for effectiveness and efficiency but also prioritizes resilience against potential challenges. Lastly, identifying methods that accommodate required constraints and preferences is essential for ensuring that the proposed scheduling solution aligns with operational realities and stakeholder needs. Chapter 4 details the most promising approaches identified in the literature and presents a mathematical model designed to replace the existing scheduling system.

> Chapter 4. Which solutionmethods are suitable to improve the current shift scheduling approach?
>
> ---
>
> - How can we transform the current model such that it allows for stochasticity incorporation?
> - Which solution method is suited to incorporate stochasticity?
> - Which solution method is suited to ensure robustness?

When one or more solution approaches are developed, they are tested to analyze their performance across various experimental settings. This is essential as it provides KLM with insights into the potential impacts of adopting the proposed solutions. First of all, the correct parameters should be chosen. Secondly, different experimental conditions may influence the quality of the model's output, such as varying inputs and operational constraints. By evaluating these settings, we can identify the advantages and disadvantages of each solution approach. Chapter 5 discusses the tests conducted and their results.

> Chapter 5. How do the different solution approaches perform in different experimental settings?
>
> ---
>
> - How does the solution method perform compared to the current scheduling model based on a practical dataset from KLM?
> - How does the stochastic solution method compare to the deterministic model?
> - How does the stochastic solution method handle different input set types?
> - How does the robust solution method compare to the stochastic solution method?

Once these questions are addressed, we can conclude the proposed approaches and formulate recommendations for KLM regarding potential changes to the scheduling process, including specific modifications to implement. In every research, certain aspects may not be fully explored, highlighting opportunities for further investigation. Identifying these areas is valuable from both an academic and a business perspective. Chapter 6 discusses the findings' conclusions, recommendations, limitations, and areas for future research.

Deliverables After finishing the research and answering the main research question, the following deliverables are presented:

- An alternative model formulation to improve computational performance of the staff scheduling problem.
- An improved model that incorporates stochasticity on the DoO.
- An analysis of the (dis)advantages of the variants of the model presented.

- A recommendation to KLM on which shift scheduling model to use.
- Recommendations for implementation of the shift scheduling model and for further research.

This concludes the outline of the research. In the next chapter, we present a detailed context analysis of KLM's current scheduling process, focusing on the operational environment at Schiphol Airport. First, we discuss the processes that are performed by the ground staff, after which we dive deeper into the scheduling process for the staff. This analysis provides a foundation for understanding the current challenges and identifying areas for improvement in the scheduling system.

2 Context Analysis

The goal of this chapter is to create a better understanding of the current situation. The second set of research questions is answered in this section. These evolve around the main research question:

"Which processes are part of ground operations and what is done to create schedules for these processes?"

Section 2.1 explains the luggage handling process from arrival to departure of a plane. Added upon this, Section 2.2 describes how and where in the process disruptions may occur and elaborates on its consequences. Subsequently, Section 2.3 covers information about the scheduling procedure of the luggage handling staff, it elaborates on the different stages in which the planning is made. Lastly, Section 2.4 elaborates on the current model that is used for this.

2.1 Processes overview

Luggage services at Schiphol Airport are advanced and complex. When passengers arrive, they drop their luggage off at check-in. The luggage handling system, owned by Schiphol Airport Group, is responsible for transferring the luggage from the Check-In to the location at Airside[1] where carts pick up the luggage and transport it to the aircraft. From luggage arriving at the apron, KLM is responsible for loading the luggage into the aircraft. Gök et al. (2020) explain the aircraft turnaround process in detail, focusing on the precedence relations between operations. Figure 1 shows that luggage (off)loading can be performed simultaneously with any other operation (which reduces complexity in the planning), and as expected, only have a precedence relation towards each other (i.e., luggage first needs to get out of a plane before new luggage goes into the plane). Subsection 2.1.1 expands on the operations that are done for the baggage, while Subsection 2.1.2 focuses on the staff that performs these operations.

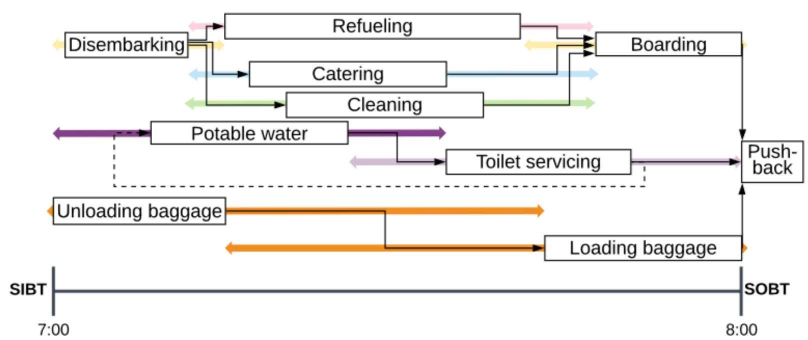


Figure 1: Aircraft turnaround operation including precedence relations (Gok et al., 2020)

2.1.1 Luggage handling

KLM organizes its luggage handling processes according to the operation type (loading or unloading) and the aircraft category. For narrow-body aircraft, the loading procedure employs a ramp snake (conveyor belt) to handle luggage in bulk, which means case by case (Figure 2a). In contrast, wide-body aircraft utilize containers to transport luggage; these containers are positioned in the aircraft in the designated hold (Figure 2b), alongside bulk-loaded luggage. In the unloading phase, the operations are reversed. Employees first identify high-priority luggage that needs to be transferred to another flight, unloading it onto a designated transfer cart. The remaining luggage is then unloaded onto a separate cart intended for luggage claim. Appendix A shows the full operations performed by the staff.

[1] Airside refers to the area where one arrives after passing through security at the airport (luggage also passes through scans before entering airside).

(a) Narrow-body

(b) Wide-body

Figure 2: Luggage loading

Baggage (un)loading must be executed and planned for every inbound and outbound flight. The operations are broken down into separate tasks, which are to be completed by a qualified employee. KLM categorizes these tasks based on several characteristics, including the start time, duration (and consequently end time), the department, and the rank required to complete the task. The deadload (weight of the luggage) is also specified, indicating the task size (used to determine task duration). Figure 3 illustrates how various tasks are structured within a specific timeframe throughout the day. Tasks that need to be performed simultaneously are stacked, allowing us to see the busiest periods of the day.

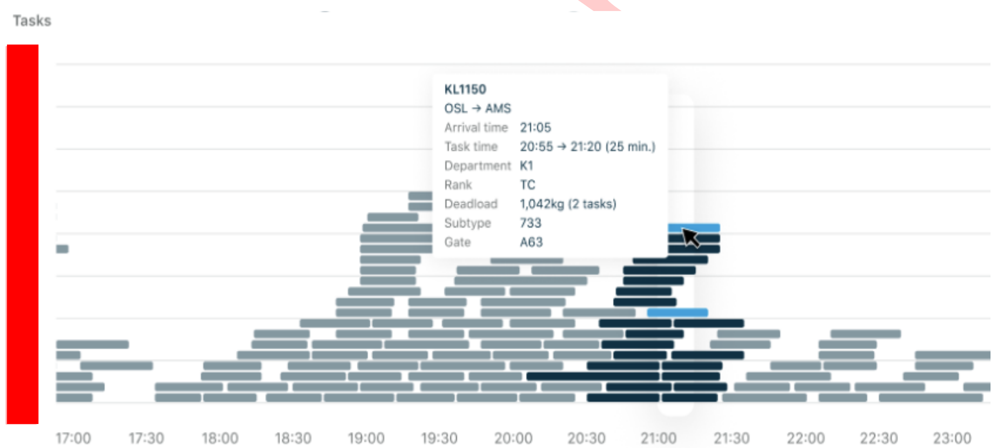


Figure 3: Example of task overview

2.1.2 Workforce context

Staff performing these tasks are not all similar in qualifications, which has consequences for the scheduling. Here, we elaborate on the most important workforce characteristics. Firstly, KLM separates their staff into three different departments. D1 represents the KLM Cityhopper (KLC), which are narrow-body aircraft within Europe, secondly, D2 typically encompasses medium-haul flights on narrow-body, but slightly larger aircraft that mainly operate within Europe. Lastly, D3 represents the intercontinental flights, which are completed with wide-body aircraft. Since the departments at the airport are far apart, it is most convenient for the staff to serve only its own department, minimizing travel times. Secondly, rank and skills are taken into consideration; ranks are Team Member (TM), Team Coordinator (TC), and Team Grand Material (TGM). The TM performs the operations as described in Section2.1.1. As the name suggests, the TC is responsible for coordinating these operations, he/she can receive information about luggage that should be loaded or kept out of the plane when necessary (i.e., if a passenger does not board the plane, or some luggage needs to be taken out quickly on arrival). The TGM is qualified to operate the luggage lift as can be seen in Figure 2b. He/she is also qualified to perform the operations of the TM (but not the other way around), whilst other mixes in qualifications are restricted. Next to the regular employees, KLM uses flex employees to cover the workload if it cannot be covered by the regular employees, which incurs higher costs and thus is not preferred. Table 1 gives an overview of the personnel resources needed. This depends on the aircraft type and the deadload that is (expected to be) on board. In our research, we focus on TM for two departments, such that the problem remains small, but exchangability can be studied.

Body	Aircraft Type	Deadload (kg)[2]	People Needed		
			TC	TM	TGM
Narrow-body	Boeing	[redacted]	[redacted]	[redacted]	[redacted]
	Boeing	[redacted]	[redacted]	[redacted]	[redacted]
	Embraer/KLC	[redacted]	[redacted]	[redacted]	[redacted]
	Embraer/KLC	[redacted]	[redacted]	[redacted]	[redacted]
Wide-body	Airbus	[redacted]	[redacted]	[redacted]	[redacted]

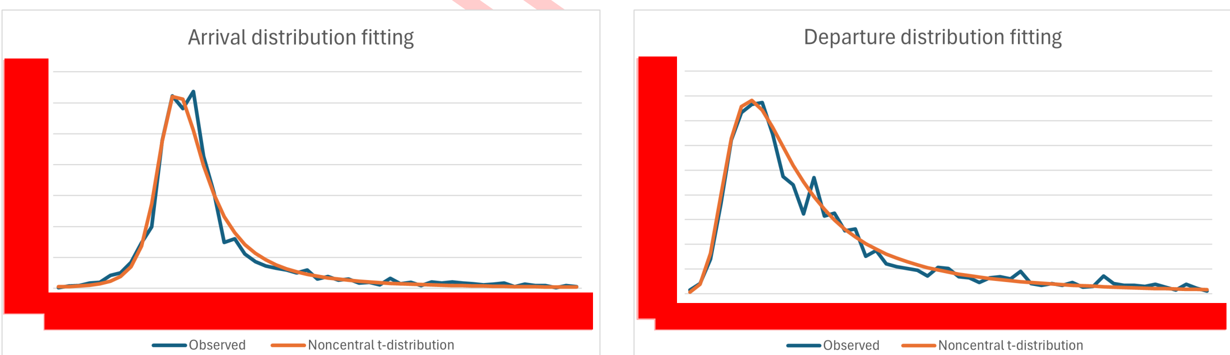
Table 1: Resources required for luggage handling by flight type

2.2 Stochasticity of day of operation

During luggage handling at the airport, not all operations proceed according to plan. Numerous factors can disrupt the schedule, leading to potential delays and inefficiencies. While many of these factors impact ground operations, only some are included in this research to maintain focus on the most critical elements. These are the most important and influential. First of all, plane delays are a major factor (Subsection 2.2.1). Seconldy, task duration deviation can influence the DoO as well (Subsection 2.2.2).

2.2.1 Plane delays

One significant source of disturbances is flight delays, as this can shift tasks to earlier or later in the day. This can create a cascading effect, pushing back the current flight's schedule and future flights that rely on the same resources. Additionally, delayed departures result in a later start time of the tasks. Arrival and departure delays are different and cannot be combined into one general 'delay' overview. We observed an average week of flight data and look into the characteristics of the delays. Figure 4 displays the observed frequency of delays accumulated over a week. It can be seen that the arrival delay peaks around [redacted] minutes, while the departure peaks at [redacted] minutes. Furthermore, the average delays are [redacted] vs [redacted] minutes, indicating that we should handle both arrival and departure deviations separately.



(a) Arrivals (b) Departures

Figure 4: Distribution fitting

Figure 4 also shows the fitted distribution: a Noncentral t-distribution for both delay categories. These are separately fitted, after which we conducted a Goodness-of-Fit test to verify the fitted these distributions. The parameters, are different and displayed in Table 2.

[2]The exact deadload that determines whether multiple members are needed is different for different sub-types but left out for simplicity.

Parameter	Arrival	Departure
Degrees of freedom	[redacted]	[redacted]
Noncentrality parameter	[redacted]	[redacted]
Location	[redacted]	[redacted]
Scale	[redacted]	[redacted]

Table 2: Distribution parameters of the fitted Noncentral t-distribution

2.2.2 Task duration

Ground operations' unloading and loading processes are inherently variable and influenced by a range of logistical, technical, and operational challenges. This variability poses significant difficulties in ensuring smooth and timely transitions, particularly during tight turnarounds at busy hubs like Schiphol Airport.

Unloading variability often stems from factors beyond the immediate control of ground staff. One of the primary challenges is the availability and functionality of equipment. Another critical issue lies in the configuration of cargo. Improperly configured pallets can impede efficient unloading, as manual adjustments or reorganization may be required. Additionally, environmental factors, such as frozen aircraft cargo doors during winter, further compound the unpredictability of the unloading process. All these possibilities are experienced as happening only very few times, and tasks are seldom finished late.

Loading operations, however, do experience delays. Loading faces its own set of challenges, often distinct from those encountered during unloading. One of the most significant issues is the dynamic nature of loading instructions. Technical problems with the aircraft may render certain cargo positions unusable, necessitating last-minute adjustments to the loading plan. Similarly, mid-process aircraft changes may cause the unloading of just loaded cargo, requiring already loaded cargo to be removed and reloaded in a new aircraft. Passenger-related issues, such as missing passengers whose luggage must be unloaded, also contribute to variability. These situations often arise with little warning, creating further delays. Since KLM wants to prevent luggage from flying to a destination without its owner, the ground staff never leaves the plane until all passengers are boarded. This causes the tasks never to be finished early, and often later.

2.3 Planning & scheduling overview

The scheduling process is divided into three distinct stages, as illustrated in Figure 5. The first stage, known as the Resource Freeze, focuses on identifying the most effective and efficient shifts for the upcoming period, this stage also evaluates whether the current flight schedule is feasible concerning long-term staffing requirements, providing insights into potential needs for hiring or training staff. Once the model is run, planners decide which shifts are necessary based on the model's output, generating what is known as the Roster Key. From here, the chosen shifts are frozen and will not be changed significantly.

Approximately three months before the DoO, a process called "P-checks" is conducted. This step provides an updated assessment of the schedule's feasibility. During P-checks, more detailed task information becomes available (e.g., flight reschedules or updated luggage forecasts). Additionally, the Roster key is used as input in the optimization model. The goal of the model is to determine whether the set shifts can cope with the demand, and after running, show whether any shortages occur. Planners then review the output to verify if the Roster key is feasible or requires adjustments, adjustments that might be needed are then accommodated, either by rearranging the planning of the permanent staff or by acquiring flex employees. In this stage of the process, planners proceed to assign individuals to shifts.

In the final stage, close to the DoO, planners have the complete information regarding permanent and flex employees, alongside updated task details. Three weeks before the DoO, the last iteration of the procedure is run, known as the Manpowered Planning System (MPS) optimization. Unlike the Resource Freeze and P-checks, which use employee rank and competency data, the MPS incorporates personal employee data, making this the most accurate forecast for staffing. The output informs planners if any last-minute adjustments are needed to ensure readiness. From this point onward, the models are no longer used.

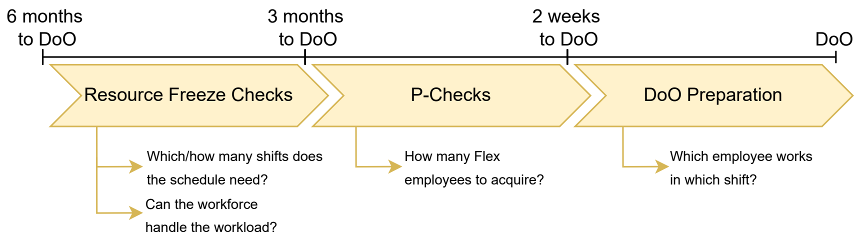


Figure 5: Planning and scheduling overview

Our research focuses on the Resource Freeze stage of the planning and scheduling process, where the flexibility to define shifts and determine staffing levels provides the opportunity to enhance schedule robustness. By incorporating robustness at this early phase, we can proactively address potential disruptions while the schedule is still adaptable, ensuring the resulting plans are both efficient and resilient. In contrast, later stages like P-checks and the Master Planning Schedule (MPS) optimization lack this flexibility, as shifts are already finalized, limiting the scope for significant adjustments. Therefore, targeting the Resource Freeze stage enables us to maximize the impact of our approach on the overall scheduling process. More departments are involved in the full personnel scheduling process, such as the flight schedule creators. Subsections 2.3.1 to 2.3.4 explain the different aspects of the to-be-constructed schedule.

2.3.1 Scheduling shifts

When deciding which shifts to use on a certain day, two decisions have to be made. First of all, the starting time and length of the shift are determined (which is defined as a 'shift-type'). The length of the shifts is usually 8.5 hours. These shifts are placed in categories (e.g. super early shift, day shift, night shift, etc), and based on these categories, the costs of the shift types are determined. The second decision to be made is the number of workers that get assigned to this shift type. This allocation must balance with the demand for tasks. Optimizing this selection process ensures an efficient distribution of staff while managing operational expenses effectively.

2.3.2 Scheduling breaks

In every shift, a break of 45 minutes should appear, this break cannot start earlier than 2 hours into the shift and no later than two hours before the end of the shift. Although breaks are flexible on the DoO and are rescheduled based on other algorithms outside of this research, they must legally be accounted for to guarantee that all employees receive their required rest periods. Each shift typically has five possible break slots, providing sufficient flexibility to integrate breaks into the schedule without disrupting workflow. Incorporating breaks into the planning model ensures compliance with labour regulations while maintaining operational efficiency.

2.3.3 Exchangability of tasks

As a starting point, all tasks are performed by employees from their skill and department, called an employee pool. On the DoO, it turns out that in some departments, understaffing is present (due to stochastic factors). Therefore, planners make use of the possibility of exchanging tasks per employee pool to satisfy demand at all times. This provides flexibility but is not preferred due to practical constraints and thus is penalized.

2.3.4 Rosterability

As planning employees comes with a lot of constraints regarding working regulations, it is necessary to design a well-balanced roster across the week. Assigning a lot of workers to the early shift category on one day and very few workers to that shift category on another day creates problems for the planners, as employees dislike a lot of irregularities and are unable to work an early shift following a late shift on the previous day. Therefore, the planners like to balance the number of shifts throughout the week. By leveling shift categories, the schedule becomes more sustainable and easier to execute. In our research, we disregard this objective since we focus on the added benefits of considering stochasticity within a day, excluding scheduling a whole week.

2.4 Current solution approach and restrictions

To schedule staff to perform operations, the scheduling team calculates the tasks that are to be performed based on the expected flight schedule. An optimization model is then applied to assign shifts to the DoO in a way that ensures all tasks are adequately covered. Figure 6 provides an example of cumulative tasks (dark blue bars) alongside the cumulative scheduled shifts (yellow curve). In this example, overstaffing is evident around 14:00, while understaffing occurs at 21:00, highlighting the importance of effective shift alignment.

Figure 6: Example of shift scheduling

The model uses three inputs. First of all, the employee pools are taken into consideration. Secondly, for each of these employee pools, the tasks that are to be performed by them are given. Thirdly, possible shifts to serve are given. Currently, it is chosen to use different shift-types based on a 15-minute time interval (i.e., shifts only start at 09:15am, 09:30am, etc.). Additionally, giving too many possibilities for the model to choose from shift types increases the model size significantly. To manage this complexity in the model, a preprocessing step is performed where ten promising shift types are selected and provided as input. For each shift type, five possible break options are evenly distributed over the shift duration, as shown in Figure 7.

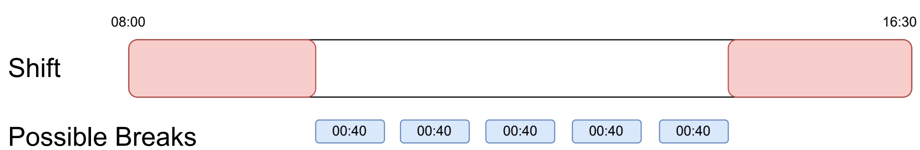


Figure 7: Possible breaks per shift

The objective of the model is to assign all tasks to shifts while determining the number of employees required for each shift. This is achieved by ensuring that the number of workers assigned to a shift matches or exceeds the number of overlapping tasks within that shift. While maintaining this constraint, the model minimizes costs, which consist of three main components. Task costs reflect whether a task is assigned to the appropriate employee pool, with higher costs applied in cases of rank or department mismatches. The second cost component is the number of shifts required, as each additional shift incurs additional expenses. Lastly, violation costs penalize uneven distribution of shifts throughout the week to promote rosterability in scheduling.

As described in Section 2.3.1, the shift scheduling (performed in the Resource Freeze Checks) decides two important things. First, the shifts that will be carried out, and second, the number of employees that will be assigned to these shifts. KLM used to optimize these two parts combined, but the runtime of the algorithm became too large (around 16 hours). Therefore, the algorithm is split into two parts: narrowing down the shift types that might be good for a schedule, and subsequently picking the best shift types and assigning employee numbers to these types. This compromises the potential of efficient shifts to use but reduces the runtime of the algorithm.

The first goal is achieved by the Shift Type Optimizer (STO), which defines the shift types to be chosen. This algorithm runs for approximately 5 minutes. The model receives all possible shifts as input (starting every half hour on a day, with the earliest start being 05:30 and the latest end being 23:59 each day, resulting in 21 possible shifts) and narrows it down to the 10 most promising shifts. Then, the Shift Optimizer (SO) is run to define which of the 10 shifts are chosen and how many employees are assigned to these shifts. The algorithm takes around 30 minutes, making the total runtime of the algorithm for the Resource Freeze Checks approximately 35 minutes.

2.5 Summary & conclusion

In this chapter, we investigated the practices at Schiphol Airport to answer the research question: *"Which processes are part of the ground operations and what is done to create schedules for these processes?"* To answer this question, we investigated the turnaround operations and identified in which field our scheduling methods could apply. As we already defined in Chapter 1, we focus on luggage handling for simplicity, Figure 1 indicates that the (un)loading operations are a standalone operation and do not require any preliminary operations. This makes it simpler and thus suitable for this research. Subsequently, it can be extended to schedule other ground-handling staff. The operations are performed by three employee groups that have different skills, and at three departments, together these form an employee pool (and thus nine pools exist).

In this research, we incorporate the stochastic nature of plane delays, as these significantly impact scheduling. Plane delays lead to task shifts and must be accounted for in the rostering process. We identified a Noncentral t-distribution to fit the plane delays and can use this to generate scenarios for our optimization model. For task durations, however, the variability is less well understood. We experiment with different variance settings for task durations to address this. This approach provides insights into how task duration variability affects scheduling outcomes. By testing these variations, KLM can determine whether collecting accurate data on task duration variability would improve the model's performance and reliability.

The planning consists of three phases, of which only in the first stage, the Resource Freeze Checks, the desired shift, and the number of those shifts are determined. In subsequent stages, checks are done to evaluate the roster, but no significant changes are made. Therefore, our research focuses on the model used in the first stage. The model used for this decides at what time a shift should start, and at what time the break should take place to minimize costs but still satisfy all tasks to be done. This goal is split up into two models. First, the possible shifts are narrowed down to some promising shifts, after which a second model decides which shifts and their number to use. This is a compromise in quality of solution but improves runtime to a reasonable length. This is all done in a deterministic setting.

In Chapter 3, we review the literature to find existing solution methods to the problem posed in this chapter, focusing on methods that are able to cope with the stochastic nature of the airport. We review and compare papers with our research to understand the similarities and differences, and subsequently identify the existing gap in literature.

3 Literature Review

The literature review investigates robust staff scheduling from a theoretical perspective. The goal of the study is to find more insights into the concept of robust staff scheduling. The following research question is answered in the following chapter:

'What methods for robust staff scheduling are discussed in the literature?'

To answer this question, first, we discuss how literature defines the problem as explained in Chapter 2 (Section 3.1). Secondly, Section 3.2 covers solution methods that are currently used to solve the problem in a traditional setting. We elaborate on the classic approaches and explain why these cannot be used in our problem. Then, we turn towards stochastic approaches for robust scheduling (Section 3.3). First, we discuss which approaches exist in the aviation domain and identify why we cannot directly use these approaches. As a solution, we look into methods that are used in other fields and determine how we can transform this into the aviation domain. In Section 3.4 we identify the gap in the literature. Lastly, we conclude the literature review by describing the promising approaches in Section 3.5.

3.1 Positioning our research in literature

In staff scheduling, ensuring workforce availability to meet demand on specific days is a well-explored topic across multiple industries. Common areas of investigation include nurse scheduling and transportation staff scheduling. As van den Bergh et al. (2013) highlights, airline operations rank as the sixth most studied field in this context, suggesting there is relevant literature for our research. Moreover, Ernst et al. (2004) note that the industries experience different requirements, making it challenging to adapt solution methods from one industry to aviation.

The staff scheduling process involves several distinct phases, as outlined by Ernst et al. (2004), who categorize it into six modules, including demand modelling, shift scheduling, and task assignment. While optimizing all modules simultaneously is desirable, doing so is computationally impractical and often misaligned with specific business needs. For example, at KLM, demand prediction precedes shift scheduling (as explained in Section 2.3), while task assignment is performed on the DoO, rather than during the shift scheduling phase. Decomposing the problem seems appropriate since KLM already imposes computational restrictions. This implies that our problem can be narrowed down to one of the six steps: shift scheduling. Shift scheduling deals with the problem of selecting, from a potentially large pool of candidates, what shifts are to be worked, together with an assignment of the number of employees to each shift. Di Gaspero et al. (2007) describe a Minimal Shift Design (MSD) problem, indicating that only the optimal shift types are determined, but the number of employees assigned to the shifts is left out. Bechtold and Jacobs (1990) were the first to explicitly model breaks of staff in the MSD problem, transforming it to a Shift-and-Break scheduling problem. This aligns with our problem best, as demand is variable, and thus efficiently placing breaks can significantly impact the schedule (Aykin, 1996; Di Gaspero et al., 2013).

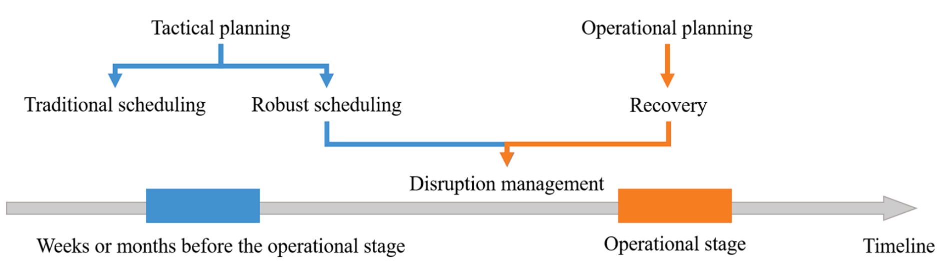


Figure 8: Categories of airline scheduling problems (Wen et al. 2021)

Wen et al. (2021) describe different stages and categories of airline scheduling problems by splitting the tactical and operational planning (Figure 8). Shift-and-Break Scheduling is classically (and at KLM) performed weeks or months before the DoO, indicating that (in contrast with for example task assignment), the problem falls under the tactical planning domain. Wen et al. (2021) state that due to the highly volatile operating environment, airlines have started aiming for robust scheduling compared to the traditional cost-minimization in tactical planning. The operational perspective deals with disrupted operations through real-time recovery solutions. We aim to transform the traditional scheduling to be resilient in the described volatile environment and therefore find ourselves in the tactical planning situation described by Wen et al. (2021), where the transformation from traditional scheduling to robust scheduling must be made.

3.2 Deterministic scheduling: state-of-the-art

The deterministic staff scheduling problems, staff scheduling problems that all knowledge is known and not subject to change, are solved in all domains using different methods. Solution methods vary from exact methods like Mixed-integer programs to heuristics approximating good solutions (van den Bergh et al., 2013). This Section covers the different solution methods applied in deterministic staff scheduling. The advantages, disadvantages, and possible usefulness of our research are stated. We start with the use of Mixed Integer Linear Program (MIP) models (Subsection 3.2.1). We then continue with papers that add extensions to the MIP (Subsection 3.2.2). Lastly, we discuss heuristics as more time-efficient algorithms in Subsection 3.2.3.

3.2.1 Mixed Integer Programming

The use of MIP for optimizing staff scheduling was first introduced by Dantzig et al. (1954), who solved a small and simplified version of the problem. Since then, significant efforts have been made by researchers to generalize the problem, extend the solution techniques to handle larger instances, and incorporate additional constraints into the MIP framework, such as breaks. Currently, KLM schedules its shifts and breaks with a MIP based on the paper by Akkermans et al. (2021). Here the problem is decomposed into two phases. In the first problem, shifts are scheduled using the ILP, while breaks are considered heuristically. In the second phase, an ILP is iteratively used to change breaks for the shift of a single employee while assuming the shifts and breaks of other employees remain fixed. This allows for numerous small ILPs, rather than one large ILP. This speeds the model up significantly but disallows joint optimization (both shifts and breaks optimized simultaneously). As a consequence, the solution found is not 'optimal'.

Hur et al. (2019a) investigated how important it is to schedule breaks together with the shifts or to leave the breaks out and assign them on the day itself, called flexible break scheduling. For this, a large break set and multiple variants of possible breaks are used, making the problem rather large and difficult to solve to optimality. The paper shows that, with randomized demand, the number of uncovered tasks can be reduced up to 94% by using flexible break scheduling. The authors use an integrated Shift-and-Break model for this, and tweak their created shift schedule in real-time using a rolling horizon. The main disadvantage of a rolling horizon approach is that the optimization model focuses on each time slice independently, missing the opportunity to perform a global optimization across all time slices.

Wang et al. (2023) show the potential to improving the shift-and-break scheduling problem by solving both the shift scheduling problem and the task assignment problem simultaneously. This is done using the 'Integrated Shift Rostering and Task Assignment' (ISRTA) model, which is innovative in that it reduces the number of variables and constraints by simultaneously assigning staff and tasks to shifts. Two methods are presented to enhance the solution process. First, the rolling horizon approach divides the problem into overlapping time slices. The algorithm solves the ISTRA model for the initial time slice, establishing optimal staff assignments, and then proceeds to the next slice, using the output from the previous time slice as input for the current one, thus improving computational efficiency and solution quality throughout the scheduling horizon. Secondly, an iterative shift selection algorithm is exploited. It enhances the model by selecting beneficial shifts while managing computational complexity. Initially, a set of shifts is generated using a larger shift start time interval, which allows for the identification of a diverse range of potential shifts. After solving the model with this initial set, the algorithm refines the shift selection by focusing on the most promising shifts from the first solution. The new ISRTA model computationally outperforms the existing separate methods for Shift Rostering and Task Assignment, with an average computational time reduction of 38.49%.

3.2.2 MIP improvements

Contributions such as Hur et al. (2019a) and Wang et al. (2023) provide high-quality solutions, but these are impractical as the running times of the algorithms used become too long for real-life instances. Therefore, methods to speed up the algorithm are constructed. In this section, we discuss methods that are used to improve shift scheduling problems in general. No specific methods are presented for the Shift-and-Break scheduling problem.

In shift scheduling, column generation as introduced by Caprara et al. (2003) is a common approach. This approach starts by solving a restricted master problem, which includes only a subset of variables (columns). A pricing sub-problem is then solved to find new columns with negative reduced costs that could improve the solution. These columns are added to the master problem, and the process is repeated until no further improvement can be made. This iterative method ensures computational efficiency when tackling large-scale issues.

Building upon column generation, Beliën and Demeulemeester (2008) propose the branch-and-price approach for solving shift scheduling problems in healthcare (excluding breaks). In Branch-and-Price, the problem is decomposed into a master problem (typically an LP relaxation) and sub-problems. After solving the master problem and generating columns through the sub-problems, a branching decision is made to ensure integrality on the relaxed variables. This method continues iteratively until an optimal solution is found or all branches are explored, allowing the handling of more complex and larger scheduling problems. Nowadays, commercial solvers use these techniques to solve optimization problems, indulging that we can solve our problem without constructing the Branch-and-Price algorithm ourselves, but using the built-in algorithms of the solvers. Cappanera et al. (2024) take a different approach to speeding up the search for solutions. A MIP formulation can be changed such that solving the problem proves easier. This is done by introducing valid inequality constraints to the problem formulation. They demonstrate that smart additions of constraints can improve the solution speed of the model.

Kletzander and Musliu (2019) evaluate three approaches to solving the shift scheduling Problem: the Direct Model, the Counting Model, and the Network Flow Model. The Direct Model uses a standard MIP to represent shifts directly by assigning employees to specific start times, shift lengths, and days (three dimensions). The Counting Model reduces the dimensional complexity of the problem by focusing on the remaining shift lengths for each period (two dimensions). Finally, the Network Flow Model treats the problem as a min-cost max-flow problem, where timeslots are represented as nodes and feasible shifts as arcs between nodes, with flow variables indicating the number of employees assigned to each shift. The network flow model outperforms both the direct and counting models, solving all instances to optimality within 15 seconds. In contrast, the direct and counting formulations struggle to find feasible solutions for some instances even with a 3600-second runtime.

3.2.3 Heuristics

Another well-known method for solving combinatorial optimization problems is by using heuristics to find good solutions. Where the use of a MIP ensures optimality (or provides a gap to optimality), such a guarantee is not present in heuristics. However, runtime performance is much higher, indicating heuristics to be a good solution to large-scale problems. Teymouri et al. (2023) investigated the use of an Adaptive Large Neighbourhood Search (ALNS) to schedule combined aircraft tail number and aircraft crew planning. The ALNS uses a (moderately good) solution to the problem and iteratively improves based on smart changes made. The MIP initially outperforms the ALNS. When instances increase, the ALNS starts to find better solutions than the MIP and does so within a sixth of the time needed. This shows that although most combinatorial optimization is solved by MIP, heuristics such as ALNS should not be discarded immediately. It should be noted that even though the research by Teymouri et al. (2023) is placed in the aviation domain, the context of the problem is different than in our research as it is an employee assignment problem rather than a shift-and-break scheduling problem.

Maenhout and Vanhoucke (2010) propose a scatter search meta-heuristic to tackle a similar crew scheduling problem. This heuristic begins by using constructive methods to generate an initial solution and explore the solution space. The meta-heuristic then refines the search, concentrating on promising areas. In this problem, duties last several hours, and each crew member can only be assigned one duty per period, making it a staff assignment problem rather than a shift scheduling problem. Since a crew member's roster is modelled as a network of duties, transforming it into a shift scheduling problem could be too time-consuming due to the growing number of duties. However, the approach is promising, as the meta-heuristic yields better solutions in 2 minutes than the exact method in up to 20 hours.

Focussing on the shift scheduling problem specifically, some heuristics are used to solve the problem, in these heuristics, breaks are disregarded. Thompson (1996) applies a simulated annealing approach to the problem, here, undercoverage (i.e., staff shortages) is disallowed. Musliu et al. (2004) is the first to include minimizing the number of shift types. The paper shows the benefits of heuristics and describes a local search approach, based on tabu search, that minimized the number of shifts active, while also minimizing under and over-coverage. Di Gaspero et al. (2007) continue on the work of Musliu et al. (2004), but simplify the problem formulation, allowing the problem to be described as a minimum edge-cost flow (MECF) problem. The advantage of such a formulation is efficient solution methods used in MECF problems, disadvantage is that assumptions need to be made, such as a non-cyclic horizon (each day is optimized separately), and the existence of an optimal solution.

Lusby et al. (2016) use the Benders decomposition method to solve a variant of the shift design problem, one where the number of shift types is minimized. In the master problem, a set of shift types is used as input, and the number of employees working each shift is solved using a MIP. The dual values of constraints are used to reveal which shift types are to be added or deleted, turning the algorithm into an iterative process of selecting better shift types and employee numbers per shift. The authors show that in real-life instances based on Copenhagen airport, the new model outperforms a classical MIP solver by 40% after 10 minutes of runtime, decreasing to 25% as runtime increases.

Kletzander and Musliu (2024) present a review of twenty hyperheuristics for the shift scheduling problem (excluding breaks). The paper uses the same objectives present in KLM: minimizing overstaffing, understaffing, and the number of shift types. The hyperheuristic uses a local search method of destroy and repair heuristics such as removing a shift type or day entirely and repairing by only using existing shift types or introducing new ones. The research discusses twenty possible approaches for selecting the destroy and repair heuristics and provides the gap to the optimal solution.

The top performers are Lean Generalized Iterated Hyper-Heuristic (L-GIHH) and Lare State Reinforcement Learning (LAST-RL). L-GIHH uses an adaptive dynamic mechanism to monitor the effectiveness of all destroy and repair heuristics and identifies the most effective pairs. LAST-RL applies reinforcement learning with a state representation based on 15 features, including the last heuristic used, its success, total improvement, and time. The authors also evaluate ALNS, as described by Teymouri et al. (2023), but demonstrate that it yields worse results, which indicates that L-GIHH and LAST-RL are more promising for our research.

3.3 Stochastic scheduling approaches: state-of-the-art

Whilst traditional scheduling problems are widely researched, variants incorporating stochasticity to achieve robustness receive much less attention. Abernathy et al. (1973) and Burke et al. (2004) explain that incorporating uncertainty is challenging due to the hierarchical personnel planning process. This process spans three phases: strategic staffing, where long-term decisions about personnel mix and budget are made to meet service demand; tactical scheduling, which aims to develop a cost-effective, high-quality baseline roster for a medium-term period based on projected service demand and staff availability; and operational allocation, where short-term variability can disrupt these medium-term predictions, leading to mismatches in actual service demand and employee availability. van den Bergh et al. (2013) define robustness as accounting for and being resilient against these mismatches. They describe three main sources of uncertainty that occur in personnel scheduling problems: uncertainty of demand (how much work there is to be done), uncertainty of arrival (when the work occurs), and uncertainty of capacity are the most critical factors, while other factors account for only minor changes. Wu (2005) claims that these disruptions are stochastic by nature and unable to be changed, so schedules must adapt to these situations. Constructing such resilient schedules involves a trade-off between disruption costs incurred by repairing the disruptions and the costs of achieving roster robustness (Wickert, 2019). As the situation changes from traditional to robust scheduling problems, other solution approaches are used (explained in Subsection 3.3.1). The approaches involve stochastic programming (similar to MIP), these are explained in Subsection 3.3.2. Furthermore, stochastic heuristics (similar to heuristics) are used and explained in Subsection 3.3.3.

3.3.1 Robust scheduling

Robust scheduling involves quantifying possible deviation dangers, a solution approach can minimize on these dangers to maximize robustness. Such a solution is proposed by Wickert et al. (2021), who define horizontal and vertical robustness as strategies to enhance system resilience. Horizontal robustness involves extending existing workforce shifts, resulting in overtime, while vertical robustness incorporates a capacity buffer to manage fluctuations. The paper introduces several metrics for assessing robustness, focusing on vertical robustness through a variable representing how many reserve shifts are assigned to an employee. This approach is extended to multi-skilled employees, allowing robustness to be measured across different skills. To ensure robustness in scheduling, the paper recommends constraining the reserve shift variable to a desired level in staff planning. The outcome of the “robust” schedule is then evaluated by calculating expected re-rostering costs based on various characteristics defined by this variable. Aloulou et al. (2013) use a horizontal method to incorporate robustness into an aircraft and passenger connection schedule. The authors allocate slack to vulnerable connections, which in our case could represent tasks that can be carried out by only a few people or tasks that are very close together. Both Wickert et al. (2021) and Aloulou et al. (2013) state that increasing the reserve shifts and slack variables decreases usefulness, as the costs of achieving this schedule are higher than a disrupted schedule, highlighting the trade-off that is present.

van Hulst et al. (2017) propose a form of robust scheduling to the shift scheduling problem that aims to optimize a shift schedule by preparing for various potential variations in demand. Rather than relying solely on expected values, this approach considers an uncertainty set, which includes possible deviations from the expected workload. The solving approach follows an iterative process: it creates an initial schedule based on expected data and subsequently the schedule is tested against the worst-case scenario from the uncertainty set. Based on the results, the model recalculates and adjusts the schedule to account for this scenario. This process of creating an initial schedule, testing the scenarios, and making adjustments continues until the schedule becomes stable across a range of scenarios, meeting a predetermined robustness tolerance. The approach outperforms the nominal approach on every test set, decreasing the amount of underutilization by 93.5% while the average objective value decreases by 50%. The approach takes more time to solve but reaches optimal values within 40 minutes.

3.3.2 Stochastic programming

Stochastic programming models the parameters of an MIP as scenarios with the objective to minimize over all these possible scenarios and thus to generalize as well as possible. Hur et al. (2019b) uses this concept by advancing on their model presented in Hur et al. (2019a). They account for stochastic demand, where tasks may need to be completed at different times. They show that a stochastic programming approach could tackle the shift and break scheduling problem. However, to make the problem solvable, the multi-stage nature of the problem was simplified into a two-stage program (i.e., this means that there is only one stage before the DoO, where we are unaware of the actual demand, and a second stage, where all demand is known. In the multi-stage program, the model considers different moments during the DoO), which reduces the complexity but sacrifices some of the nuances of the real-world scenario.

Poullet and Parmentier (2020) address shift scheduling under uncertainty by incorporating stochasticity, assuming that unassigned tasks are outsourced at a high cost. The model frames the problem as a stochastic path problem, where each path represents a potential shift schedule that includes tasks and breaks. Random variables associated with each arc account for variability in job completion times and delays, to minimize the expected cost across all scenarios. Using column generation, the authors start with a simple master schedule, iteratively introducing new shifts to improve the relaxed master problem by leveraging dual constraint values. This approach yielded near-optimal solutions for up to 405 jobs across 100 scenarios. The authors benchmarked the solution with a deterministic version. This showed potential cost savings of 3.5% to 4.8%, suggesting even greater savings as instance sizes grow and reinforcing the motivation for further research.

Wu et al. (2023) address the shift scheduling problem by converting a deterministic MIP model into a stochastic MIP. This is achieved by reformulating the model to focus on the probability of meeting staff requirements, rather than simply satisfying a demand vector. They use the Sample Average Approximation (SAA) method to estimate the expected outcome of the stochastic model, calculating an average objective based on a set of demand samples generated from the demand's probability distribution. As the scenario set grows, they enhance solution efficiency through a two-stage heuristic (TSH). The first stage of TSH constructs a demand curve using a greedy algorithm. In contrast, the second stage utilizes this staffing curve as input for a simpler shift design model, aiming to minimize staffing and overstaffing costs, without considering understaffing. Although the SAA method yields the most favourable results, the TSH method approaches these results when the scenario sample size reaches 10,000, with each time slot being one hour and a runtime limit of 20 minutes.

So far, only general demand stochasticity is taken into consideration, which can be a consequence of task completion time deviations or plane delays, but is left unspecified. Liu et al. (2020) look specifically into the consequences of plane delays in the task assignment problem at airport check-in counters. The authors add a set of scenarios corresponding to the task starting times, which subsequently changes the demand requirements for each period. Then, a risk-averse model is proposed, that penalizes large deviations more extremely than light deviations and the model is solved using SAA and the Progressive Hedging Algorithm (PHA). PHA operates by solving separate subproblems for each scenario iteratively, updating costs based on previous solutions to improve convergence. The results indicate that while SAA might be quicker for small problems, PHA's ability to handle larger and more complex problems makes it a valuable alternative, especially when facing significant uncertainties in scheduling.

Ingels and Maenhout (2019) are the first to incorporate capacity uncertainty into a personnel scheduling problem by allocating capacity buffers. Their model transforms the stochastic MIP into a deterministic one by defining a range for staffing requirements rather than a fixed number. Using a buffer budget, the model strategically allocates these buffers across time slots while remaining within defined limits. This approach enables dynamic staffing adjustments that anticipate demand fluctuations, optimally positioning buffers across shifts and days. The model enhances resilience by allowing rosters to adapt to potential over- or under-demand scenarios, balancing stability and cost-effectiveness. As a result, the authors observed higher stability, i.e., a smaller number of shortages, re-assignments and cancellations.

3.3.3 Heuristics

As described in Section 3.2.3, heuristics can be a solution for large-scale problems. So far, all stochastic solutions presented in the literature review involve a stochastic MIP, which will likely result in infeasibly long solving times. Therefore, heuristics are likely to be a solution to the problem. Juan et al. (2021) present different types of heuristics and their role in stochastic optimization. Simulation-based optimization is mentioned as the most suitable approach. This approach combines heuristics with simulation methods like Monte Carlo simulation. Learnheuristics, which make use of Machine-Learning methods to learn the dynamics of input parameters and try to predict them, are an alternative, but less suitable as they focus on online changes of a (scheduling) problem.

Helber and Henken (2010) approach the shift design problem under uncertainty in call centres. They incorporated the stochasticity of arrivals and processing times into the simulation-optimization approach by optimizing over a set of different scenarios to generate shift schedules with average profit robust to changes in call arrival and processing.

Gök et al. (2020) apply simulation-optimization to apron turnaround operations, modelling several stochastic components such as the arrival times of aircraft, processing times for turnaround tasks, team travel times between consecutive turnarounds, and resource replenishment times for finite capacity resources. The approach begins with a constructive “simple” algorithm to generate an initial schedule, which is then refined using a Large Neighbourhood Search (LNS) that maximizes slack between operations and thus enhances resilience. A simulation is run to evaluate the solution, and if performance thresholds are not met, another LNS iteration is performed. The results show that simulation optimization significantly reduces delays, although it struggles to find solutions within an “acceptable” runtime when variability in the uncertainty factors is high. Gök et al. (2023) advance on their earlier work by introducing a more advanced feedback mechanism to make specific improvements to the schedule. These constraints force higher slack between certain tasks, improving the resilience.

Liu et al. (2024) use a similar approach to Gök et al. (2020) but introduce a more advanced feedback mechanism to solve a project scheduling problem (which is significantly different from our shift scheduling problem). Here an advanced conflict resolution mechanism is used. Unlike the model by Gök et al. (2023), who specifically restrict the minimal slack, Liu’s model iteratively refines the schedule across separate windows, which allows the scheduling horizon into smaller, manageable intervals. The authors describe that the framework also uses a matching algorithm to differentiate hard from soft constraints, allowing the optimization to first satisfy critical resource requirements before addressing more flexible demands. The advanced feedback given by the simulation model into the optimization model and the separation into windows allows the model to search for promising solutions while keeping the runtime acceptable. This could be an interesting approach to our minimal shift design problem, where randomly searching for better solutions likely leads to solutions that will not improve the solution.

3.4 Contribution Statement

Despite the extensive study of the deterministic Shift-and-Break scheduling problem, the stochastic variant remains relatively underexplored. Our contribution fills this gap by addressing key dimensions lacking in existing literature, as highlighted by the comparative summary in Table 3. In the table, we provide an in-depth comparison of specific problem characteristics, objectives, and solution methodologies from six studies most closely related to our work. We included papers specifically addressing the shift scheduling problem, excluding those that tackle closely related problems. Additionally, only the most recent studies incorporating stochasticity and the most promising deterministic study, the Network Flow Model from Kletzander and Musliu (2024), are selected. The model by Kletzander and Musliu (2024) shows that a new formulation can achieve the same results while being computationally superior to the MIP approach.

	Our research	Hur et al. (2019b)	van Hulst et al. (2017)	Wu et al. (2023)	Poullet and Parmentier (2020)	Ingels and Maenhout (2019)	Kletzander and Musliu (2024)
Model type							
Shift creation	✓	✓	✓	✓	✓		✓
Employee assignment						✓	
Task assignment					✓		
Constraints							
Part-time shifts	✓	✓		✓	✓		
Multi-skilled workforce	✓						
Breaks included	✓	✓			✓		
Consecutive days working					✓	✓	
Exchangeability	✓						
Objective							
Number of shifts	✓	✓	✓	✓	✓	✓	✓
Different shift types			✓			✓	
Outsourcing					✓		
Overstaffing	✓		✓	✓			✓
Understaffing	✓	✓	✓	✓		✓	✓
Exchangeability	✓						
Stochasticity							
Demand-uncertainty	✓	✓	✓	✓	✓	✓	
Capacity-uncertainty						✓	
Solution Approach							
	Stochastic MILP, SAA	MILP, Column generation	Nonlinear knapsack	Stochastic MILP, SAA	MILP, Column generation	MILP, simulation optimization	Network Flow Model

Table 3: Comparison of Solution Methods and Incorporated Constraints across Papers

From this comparison, we identify Hur et al. (2019b) and Poullet and Parmentier (2020) as particularly relevant due to their inclusion of breaks in shift scheduling. Hur et al. (2019b) shares several objectives with our study but does not account for shift-type costs or overstaffing penalties. Instead, it emphasizes the benefits of scheduling breaks later in a shift. However, this model is not suitable to solve real-life instances by incorporating break scheduling. Their approach simplifies demand uncertainty by adjusting workforce requirements per scenario, without explicitly modeling task delays. Moreover, they do not consider the trade-off between shift costs and robustness, nor do they account for workforce heterogeneity—where employees possess different skill levels and task exchanges between skill levels may not always be bidirectional. Wu et al. (2023) can solve a stochastic version using the SAA, but neglects the scheduling of different breaks, making its model distant from the KLM case.

Our work extends the shift scheduling problem by incorporating both shift types and workforce size while accounting for a multi-skilled workforce capable of task exchange to meet demand. Additionally, we model this problem in a stochastic environment, where workforce demand fluctuates throughout the day. By formulating the model to handle real-life instances, such as those encountered by KLM, we enhance its practical applicability.

3.5 Summary & conclusion

In this chapter, we researched the literature to answer the research question: *"What methods for robust staff scheduling are discussed in the literature?"* To answer this question, we first narrowed the problem down to a shift-and-break scheduling problem in the tactical domain, focusing on robust scheduling (incorporating stochasticity) rather than on traditional scheduling methods. We first researched the different state-of-the-art methods in the deterministic setting since this is more widely researched. Hur et al. (2019a) describe the importance of not scheduling the breaks of ground personnel until the DoO. Secondly, Kletzander and Musliu (2019) proposed that formulating the problem as a Network Flow model is computationally efficient. We discovered the use of heuristics in case the exact approaches are not sufficient after all. Most promising are the heuristics first introduced by Musliu et al. (2004) and later expanded upon by Kletzander and Musliu (2024), who describe to use Hyper heuristics to solve the problem. In this approach, the local search operators of Musliu et al. (2004) are chosen by the advanced operator selection algorithm of Kletzander and Musliu (2024).

In a stochastic setting, Wu et al. (2023) identified in a structured manner how the deterministic variant can be transformed into a stochastic model and promisingly presented SAA as a solution method. The downside of using this approach is that it is computationally difficult. Poullet and Parmentier (2020) and Hur et al. (2019b) tackle this by decomposing the problem and using column generation. In addition, both papers describe how to include breaks into the scheduling, which is essential for KLM. This model can still be too computationally demanding to be solved for KLM. Therefore, we looked into how the heuristics are used in stochastic settings. Juan et al. (2021) shows that simulation-based optimization is a promising approach. Combining the hyperheuristics of Kletzander and Musliu (2024) into this simulation-based optimization framework might result in better computational performance, and still achieve good solutions.

In the next chapter, we will elaborate on our own solution approach, which expands on the existing literature by incorporating exchangability. Furthermore, we improve the computational performance by efficiently reformulating the current solution method by KLM. We first explain how we transform the solution approach, after which we present multiple variants of a stochastic solutions.

4 Model Formulation

In this chapter, we introduce a shift scheduling model designed to incorporate robustness in a stochastic framework. The central research question guiding this chapter is:

'What methods can effectively integrate robustness into the current shift scheduling approach?'

Section 4.1 provides an overview of the current modeling approach alongside an alternative method aimed at improving model efficiency. We evaluate these methods by comparing their scale and anticipated performance, highlighting the strengths and weaknesses. In Section 4.2, we present the detailed formulation of the proposed robust scheduling model. Lastly, Section 4.3 presents a model formulation where robustness guarantees are modeled.

4.1 Comparing current and alternative approaches

This section compares the current task assignment-based approach (Subsection 4.1.1) with an alternative method (Subsection 4.1.2) designed to address shift scheduling challenges. The current approach, detailed in Section 2.4, minimizes the number of employees required by solving a task assignment problem. In contrast, literature often tackles such problems by aggregating demand over multiple time periods. Here, we examine the distinctions between these models (Subsection 4.1.3), evaluate their respective strengths and weaknesses by formulating an evaluation model (Subsection 4.1.4), and identify which approaches are most suitable for adapting to a stochastic framework.

4.1.1 Current approach

The current approach looks at the day by considering all separate tasks to be done. The optimization model determines the minimal number of shifts required to ensure each task can be covered. The notation of the model, referred to as the Task Assignment Model (TAM), is summarized in Table 4. We note that rosterability as explained in Subsection 2.3.4 is neglected; thus, this model can be seen as a simplified version of the model used. The first set denoted is the set of employee pools, characterized by their department and skill. Secondly. A set of tasks is given, where each task is to be performed by exactly one employee and is defined by its start time, end time, employee pool, and skill. Combining info from the first two sets, a set of tasks for each employee pool is constructed, forming multiple subsets of tasks. Additionally, there is a set called 'maximal cliques', which is constructed [redacted] Appendix B shows how this set is constructed. Finally, we define a set of possible shifts (with breaks included, making the set large), as described in Section 2.4. The parameters of the model represent various costs, such as the cost of assigning employees to shifts, the cost of exchanging tasks between employee pools, and the maximum number of tasks that can be exchanged between pools. The variables include the assignment of tasks to shifts, which is represented as a binary variable, and the number of employees assigned to each shift, which is represented as an integer variable. Together, these components form the foundation of the TAM, which is designed to optimize shift assignment effectively.

Index and set	Definition
$e \in E$	Set of employee pools
$t \in T$	Set of all tasks
$t \in T_e$	Set of tasks for employee pool e ($T_e \subseteq T$)
$c \in C$	Set of all maximal cliques
$t \in O_c$	Set of overlapping tasks in clique c
$s \in S$	Set of all possible shifts of a day
$s \in S_t$	Set of shifts that can perform task t
Parameter	**Definition**
$C_{e,s}$	Cost of assigning worker from employee pool e to shift s
$W^e_{e'}$	Cost of exchanging a task from pool e to a worker from pool e'
$V^e_{e'}$	Maximum number of exchanges to assign from pool e to pool e'
Variable	**Definition**
$x_{s,t,e'} \in \{0,1\}$	1 if task t is assigned to shift s from pool e to pool e', 0 otherwise
$w_{e,s} \in \mathbb{Z}^+$	Number of employees from pool e assigned to shift s

Table 4: Notation for the Task Assignment Model (TAM).

The TAM is defined as:

$$\min\ z = \sum_{s\in S}\sum_{e\in E}\left(w_{e,s}\cdot C_{e,s} + \sum_{t\in T_e}\sum_{e'\in E} W^e_{e'}\cdot x_{s,t,e'}\right) \tag{1}$$

s.t.

$$\sum_{s\in S_t}\sum_{e'\in E} x_{s,t,e'} = 1 \quad \forall t\in T \tag{2a}$$

$$\sum_{t\in O_c} x_{s,t,e} \leq w_{e,s} \quad \forall s\in S, c\in C, e\in E \tag{2b}$$

$$\sum_{t\in T_e} x_{s,t,e'} \leq V^e_{e'} \quad \forall e,e'\in E \tag{2c}$$

The objective function (1) minimizes two key components: the total number of employees assigned to shifts and the total cost incurred when tasks are exchanged between employee pools. Constraint (2a) ensures that each task is assigned to exactly one shift. Constraint (2b) guarantees that the number of employees assigned to a shift is sufficient to handle overlapping tasks (i.e., tasks that are to be performed at the same time) within the same shift. Finally, Constraint (2c) limits the maximum number of task exchanges between employee pools, ensuring that the model adheres to predefined exchange restrictions. As stated in Section 2.4, this model is run after the STO and determines the ideal shift types and the number of people needed for each shift type. The latter is presented by the variable $w_{e,s}$.

4.1.2 Proposed alternative

This section introduces an alternative formulation for the task assignment model. The proposed approach shifts the focus from individual task assignments to aggregating demand per period. Shifts are then scheduled to ensure that the demand in each period is met. This approach is based on the formulations used by Hur et al. (2019b) and Wu et al. (2023) who present the demand as integer variable instead of separate binary variables. We transform exchangability as presented in the TAM ourselves to comply with the new formulation and extend to a stochastic version this model later on in this research (Section 4.2). We refer to the model as the Demand Accumulation Model (DAM). In Section 4.1.3 we show that applying an aggregated demand formulation results in a significantly reduced model, improving computational performance.

Transformation from current approach In this alternative approach, we do not approach tasks separately, but we discretize the planning horizon into separate periods and accumulate the tasks within these periods to represent the total demand in this period. This means that a task that spans two periods is treated as two separate tasks, one for each period. While this simplification helps in constructing the model, it introduces challenges, particularly when shifts or breaks start or end. In the original model formulation, it was not allowed to perform a task that spans longer than the shift of an employee, whilst, in our new approach, it is possible to perform only one period of this task, whilst another employee takes care of the second part. This can be seen as if a task is transferred between multiple employees. Figure 9 shows this phenomenon. It displays how a shift (blue lines) can end and 2 new ones can start, whilst Task 1 spans over periods 1 and 2, and thus is covered by different employees. We expect the impact to be minor, as the task durations are only a margin of the length of shifts, so this phenomenon is only expected to occur in a fraction of all cases. To evaluate the practical impact of these assumptions, a separate model is proposed, detailed in Section 4.1.4.

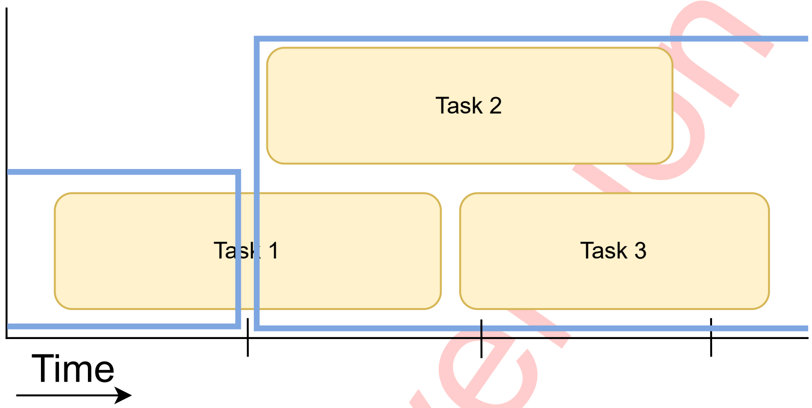


Figure 9: Task split between shifts

Constructing the demand data The demand data used in this thesis is based on task schedules provided by KLM, where tasks can start at any minute of the day. To convert this into period-based demand, we apply the following criteria.

The inclusion criterion states that if a task spans more than half of a period, it is included in that period's demand. Conversely, the exclusion criterion dictates that if a task spans less than half of a period, it is excluded from that period. Figure 10 displays how the tasks (left) are converted into a demand per time (middle), and further discretized into demand per period (right). This discretization is essential for efficient problem modeling. Without it, additional variables are required to capture all workload variations. Furthermore, it shows how minimal task deviations can lead to different demands per period, as in scenario A, task 1 spans a little over half of period 1, whilst, in scenario B, task 1 is only a little less, resulting in no demand during period 1 in scenario 1.

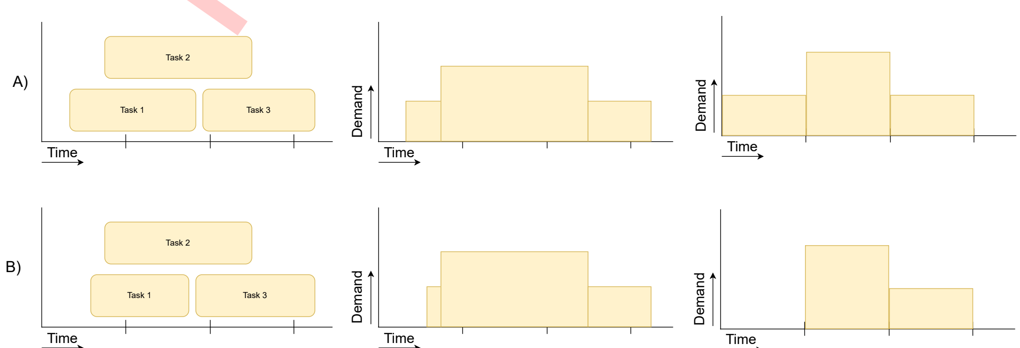


Figure 10: Task to demand conversion

We look into the task durations to identify whether this transformation results in an unfair display of reality. Figure 11 shows us that no tasks are shorter than [redacted] minutes, meaning that if the period is not chosen longer than [redacted] minutes, no task completely disappears. However, tasks might be shortened or extended slightly. This rounding approach assumes that, across many tasks, the upward and downward adjustments balance out, minimizing any significant impact on the overall model's accuracy.

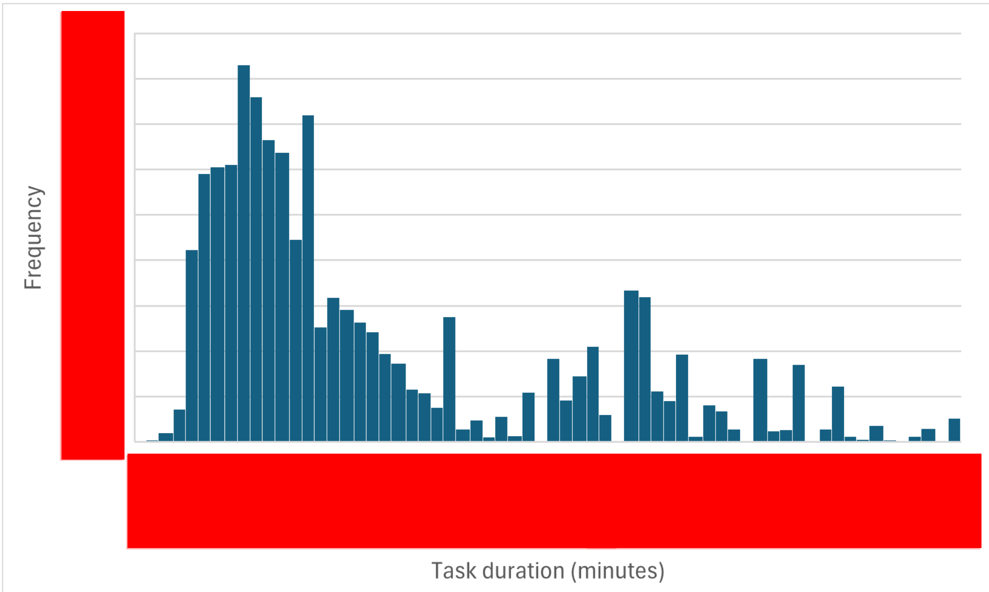


Figure 11: Task duration histogram

The following assumptions are made for shifts and breaks. Shifts are assumed to start every half hour and last for 8.5 hours, which aligns with KLM's current scheduling practices. Breaks are modelled to align precisely with period boundaries, where each break spans a whole number of periods and starts and ends at these boundaries. This alignment assumption is reasonable, as the period duration (e.g., 5 minutes) is a fraction of KLM's current break duration (45 minutes), resulting in minimal deviations introduced by periodization.

Model formulation The sets used in the proposed model include the periods in the planning horizon, the employee pools, and the possible shift breaks. The parameters in this model deviate significantly from the current approach. First, the model incorporates demand per period for each employee, as described earlier. Additionally, a binary parameter indicates whether a shift is active during a period, ensuring that workers assigned to that shift are actually working. The costs of exchanging tasks during a period and the maximum number of periods a task can be exchanged are similar to the current model. However, while the current approach focuses on satisfying all tasks and only penalizes overstaffing, the proposed model explicitly penalizes both overstaffing and understaffing (although the penalty for overstaffing is determined to be zero). This flexibility allows planners to fine-tune the model based on specific needs. The model also uses several variables. The most important is the number of workers assigned to each shift. Other variables account for the number of workers assigned during specific periods, the exchanges between employee pools, and the excess or shortfall of workers.

Index and set	Definition
$p \in P$	Set of periods
$e \in E$	Set of employee pools
$s \in S$	Set of all possible shift breaks
Parameter	**Definition**
$d_{p,e}$	Demand of employee pool e during period p
$A_{s,p} \in \{0,1\}$	Indicator whether shift s is active during period p
$W_{e,e'}$	Costs of exchanging one employee from pool e to pool e'
$V_{e,e'}$	Maximum number of workers that can be exchanged between two employee pools e and e'
U	Costs of understaffing
O	Costs of overstaffing
Variable	**Definition**
$C_{e,s}$	Cost of assigning worker from employee pool e to shift s
$x_{e,s} \in \mathbb{Z}^+$	Number of workers from employee pool e assigned to shift break s
$w_{p,e} \in \mathbb{Z}^+$	Number of workers from employee pool e working during period p
$v_{p,e,e'} \in \mathbb{Z}^+$	Number of workers from employee pool e assigned to employee pool e' during period p
$o_{p,e} \in \mathbb{Z}^+$	Overstaffing during period p for employee pool e
$u_{p,e} \in \mathbb{Z}^+$	Understaffing during period p for employee pool e

Table 5: Notation for the Periodic Shift Scheduling Model.

The model is defined as:

$$\min\ z = \sum_{s\in S}\sum_{e\in E} C_s \cdot x_{e,s} + \sum_{p\in P}\sum_{e\in E}\left(o_{p,e}\cdot O + u_{p,e}\cdot U\right) + \sum_{p\in P}\sum_{e\in E}\sum_{e'\in E} W_{e,e'}\cdot v_{p,e,e'} \tag{3}$$

s.t.

$$w_{p,e} - \left(o_{p,e} + \sum_{e'\in E} v_{p,e,e'}\right) + \left(u_{p,e} + \sum_{e'\in E} v_{p,e',e}\right) = d_{p,e} \quad \forall p \in P, e \in E \tag{4a}$$

$$w_{p,e} = \sum_{s\in S} x_{e,s}\cdot A_{s,p} \quad \forall e \in E, p \in P \tag{4b}$$

$$\sum_{e'\in E} v_{p,e,e'} \leq w_{p,e} \quad \forall e \in E, p \in P \tag{4c}$$

$$\sum_{p\in P} v_{p,e,e'} \leq V_{e,e'} \quad \forall e \in E, e' \in E \tag{4d}$$

The objective function in (3) consists of three components: the costs of assigning employees to shifts, the costs for exchanging tasks, and penalties for overstaffing and understaffing. These factors are similar to the current model, but the penalties provide more flexibility in managing workforce requirements. The constraints work together to ensure proper scheduling: Constraint 4a ensures that overstaffing and understaffing are accounted for in each employee pool and period, while Constraint 4b determines the number of employees currently working. Additionally, Constraint 4c limits the number of exchanges between employee pools, ensuring that exchanges do not exceed the number of workers assigned to shifts, and Constraint 4d sets a maximum on the total number of exchanges between employee pools, in line with the current model's limitations.

4.1.3 Model comparison

To compare the performance of the TAM and DAM, we examine the size of the models. In the current approach, two variables are used that depend on employee pools, shifts, and tasks. The key difference between the TAM and DAM is that, in the latter, the number of variables does not increase with the number of tasks but only with the number of periods. This means that as the "to be planned" task set grows, the model in the DAM is expected to maintain consistent runtime. Similarly, the constraints in the DAM show the same trend, with an additional constraint related to the number of cliques. A summary of the variables and their sizes is shown in Table 6.

To compare the performance of both models, we use the input data provided by KLM. The model involves 9 employee pools (3 departments and 3 skill levels), with 21 shifts (all possibilities), each having 5 break possibilities, totalling 105 possible shifts. Note that this number of shifts is used by KLM after running the STO, so the solution space is already narrowed down. There are approximately 3,400 tasks per day, and the number of cliques is estimated to be around 2,200 (based on multiple test runs). The model uses a 15-minute period length, with the schedule spanning from 05:30 to 23:30 each day, resulting in 72 periods. In terms of variables, the total in the current approach is 3,213,945 (numbers are scaled such that the largest number is 1,000,000), whereas in the DAM, this number is reduced by 99.7%, down to 2,713.5. Similarly, the number of constraints is reduced by 99.9%, from 647,951.7 in the TAM to 630.1 in the DAM.

Current approach		**Proposed alternative**	
Variables			
$x_{s,t,e'}$	$\lvert S\rvert \times \lvert T\rvert \times \lvert E\rvert$	$x_{e,s}$	$\lvert E\rvert \times \lvert S\rvert$
$w_{e,s}$	$\lvert E\rvert \times \lvert S\rvert$	$w_{p,e}$	$\lvert P\rvert \times \lvert E\rvert$
		v_{p,e_1,e_2}	$\lvert P\rvert \times \lvert E\rvert^2$
		$o_{p,e}$	$\lvert P\rvert \times \lvert E\rvert$
		$u_{p,e}$	$\lvert P\rvert \times \lvert E\rvert$
Total number: 1,000,000		Total number: 2,713.5	
Constraints			
2a	$\lvert T\rvert$	4a	$\lvert P\rvert \times \lvert E\rvert$
2b	$\lvert S\rvert \times \lvert C\rvert \times \lvert E\rvert$	4b	$\lvert P\rvert \times \lvert E\rvert$
2c	$\lvert E\rvert^2$	4c	$\lvert P\rvert \times \lvert E\rvert$
		4d	$\lvert E\rvert^2$
Total number: 647,951.7		Total number: 630.1	

Table 6: Variable and constraint sizes comparison

4.1.4 Evaluation model

In this Section, we propose a variation of the TAM to evaluate the performance of the DAM, this can show the impact of the aggregation of demand. The modifications are outlined here. Section 4.1.2 discusses how the solution of the DAM can lead to discrepancies between the real-world situation and the model. In this revised model, the proposed shifts are fixed as input parameters (denoted as $Y_{e,s}$), and the TAM is run to minimize the number of unassigned tasks. To achieve this, the objective function is modified to focus on the number of unassigned tasks, as described in Equation 5. When the objective value of this model equals 0, all tasks are assigned. Constraint 2a is split into two separate constraints. Constraint 6a ensures that each task can be assigned at most once, while Constraint 6b ensures that tasks are not assigned to shifts that are incapable of performing them. Additionally, Constraint 6c is introduced to fix the variables to the output of the alternative model.

$$\min\ z = |T| - \sum_{s \in S_t} \sum_{t \in T} \sum_{e' \in E} x_{s,t,e'} \tag{5}$$

s.t.

$$\sum_{s \in S_t} \sum_{e' \in E} x_{s,t,e'} \leq 1 \quad \forall t \in T \tag{6a}$$

$$\sum_{\substack{s \in S \\ s \notin S_t}} \sum_{e' \in E} x_{s,t,e'} = 0 \quad \forall t \in T \tag{6b}$$

$$w_{e,s} = Y_{e,s} \quad \forall e \in E, s \in S \tag{6c}$$

4.2 Stochastic model

In this Section, we propose a stochastic variant to the DAM. In this model, we vary the demand for each time period. the variations chosen are presented in Section 2.2. This model is referred to as the Stochastic Demand Accumulation Model (SDAM).

4.2.1 Formulation stochastic model

The formulation of the SDAM is very similar to the model proposed in Subsection 4.1.2. The differences are elaborated upon. First of all, a set of scenarios is introduced. We construct these scenarios by sampling, for each task in the input set (which is defined similarly to the TAM) a delay in start time. This delay is taken from the Noncentral t-distribution as introduced in Subsection 2.2.1. All task delays are assumed to be uncorrelated. After the sample is taken, we convert the tasks to demand as proposed in Subsection 4.1.2. Since employees are assigned to shifts before the day of operations, $x_{e,s}$ is a first-stage variable. Consequently, the number of people working in each period ($w_{p,e,s}$) is determined beforehand and is not dependent on the scenario. Since demand changes, but the availability of personnel remains equal, the over and understaffing vary and become dependent on the scenario. The only way to mitigate some over and understaffing is by changing the exchanged employees in certain periods, this variable is therefore also part of the second stage variables.

Index and set	**Definition**
$p \in P$	Set of periods
$e \in E$	Set of employee pools
$s \in S$	Set of all possible shifts
$\omega \in \Omega$	Set of scenarios
Parameter	**Definition**
$C_{e,s}$	Cost of assigning worker from employee pool e to shift s
$d^{\omega}_{p,e}$	Demand of employee pool e during period p in scenario ω
$A_{s,p} \in \{0,1\}$	Indicator whether shift s is active during period p
$W_{e,e'}$	Costs of exchanging one employee from pool e to pool e'
$V_{e,e'}$	Maximum number of workers that can be exchanged between two employee pools e and e'
U	Costs of understaffing
O	Costs of overstaffing
π^{ω}	Probability of scenario ω
First stage Variables	**Definition**
$x_{e,s} \in \mathbb{Z}^+$	Number of workers from employee pool e assigned to shift break s
$w_{p,e} \in \mathbb{Z}^+$	Number of workers from employee pool e during period p
Second stage Variables	**Definition**
$v^{\omega}_{p,e,e'} \in \mathbb{Z}^+$	Number of workers from employee pool e assigned to employee pool e' during period p in scenario ω
$o^{\omega}_{p,e} \in \mathbb{Z}^+$	Overstaffing during period p for employee pool e in scenario ω
$u^{\omega}_{p,e} \in \mathbb{Z}^+$	Understaffing during period p for employee pool e in scenario ω

Table 7: Notation for the Shift Scheduling Model.

$$\min\ z = \sum_{s\in S}\sum_{e\in E} C_{e,s}\cdot x_{e,s} + \sum_{\omega\in\Omega}\pi^{\omega}\cdot\left(\sum_{p\in P}\sum_{e\in E}(o^{\omega}_{p,e}\cdot O + u^{\omega}_{p,e}\cdot U) + \sum_{p\in P}\sum_{e_1\in E}\sum_{e_2\in E} W_{e,e'}\cdot v^{\omega}_{p,e,e'}\right) \tag{7}$$

s.t

$$w_{p,e} - \left(o^{\omega}_{p,e} + \sum_{e' \in E} v^{\omega}_{p,e,e'}\right) + \left(u^{\omega}_{p,e} + \sum_{e' \in E} v^{\omega}_{p,e',e}\right) = d^{\omega}_{p,e} \quad \forall p \in P, e \in E, \omega \in \Omega \tag{8a}$$

$$w_{p,e} = \sum_{s \in S} x_{e,s} \cdot A_{s,p} \quad \forall e \in E, p \in P \tag{8b}$$

$$\sum_{e' \in E} v^{\omega}_{p,e,e'} \leq w_{p,e} \quad \forall e \in E, p \in P, \omega \in \Omega \tag{8c}$$

$$\sum_{p \in P} v^{\omega}_{p,e,e'} \leq V_{e,e'} \quad \forall e \in E, e' \in E, \omega \in \Omega \tag{8d}$$

The objective function is changed to minimize each scenario's overstaffing, understaffing, and exchangeability. Therefore, these parts of the objective function are changed to sum over the different scenarios. The chosen shifts remain unchanged. Constraints 8a, 8c, and 8d, are changed to make sure that in each scenario, these constraints are not violated.

4.2.2 Sample Average Approximation algorithm

Section 4.1 provides a deterministic solution to a problem that is not deterministic in nature, therefore, this is rather an approximation model that uses averages. To improve the quality, we shift to a stochastic programming model and use the Sample Average Approximation (SAA), which approximates the objective value of the stochastic model by evaluating a sample of $|N|$ scenarios, replacing the full scenario set Ω. We follow this method as presented by Kleywegt et al. (2001); Ahmed and Shapiro (2002); Bentaha et al. (2014) and later used by Leeftink et al. (2019) (we stick to the notation used in Leeftink et al. (2019)). The SAA method uses the randomly drawn scenarios as described in Subsection 4.2.1. The objective value can be approximated by averaging over all scenarios N. This gives the objective function:

$$\min \ z = \sum_{s \in S} \sum_{e \in E} C_s \cdot x_{e,s} + \frac{1}{|N|} \sum_{n \in N} \cdot \left(\sum_{p \in P} \sum_{e \in E} (o^n_{p,e} \cdot O + u^n_{p,e} \cdot U) + \sum_{p \in P} \sum_{e_1 \in E} \sum_{e_2 \in E} W_{e_1,e_2} \cdot v^n_{p,e_1,e_2} \right) \tag{9}$$

The constraints used in the model are equal to constraints 8a-8d, the scenario set Ω is replaced by the scenario set N. We use $|M|$ replications of $|N|$ samples. In each replication m, random samples are generated (size $|N|$). The objective value is $\hat{v}^m_N$ with $\hat{x}^m_N$ being the corresponding solution. We use the results from different replications to identify the variation in outcome (objective value and the differences in solution).

To determine the number of scenarios sufficient to accurately estimate the 'true' objective value, we use a control scenario set N', which is significantly larger than the chosen number of scenarios N. After running the model for N scenarios and M replications, we find an estimator of the objective value $E\left[\hat{v}_{|N|}\right]$ and thus a lower bound presented by:

$$\bar{v}^{|M|}_{|N|} = \frac{1}{|M|} \sum_{m \in M} \hat{v}^m_{|N|} \tag{10}$$

After finding the lower bound, We evaluate the model's outcome by fixing the resulting shifts $\hat{x}^m_N$ and observing the corresponding objective value based on the control set N'. We can then find the upper bound of the objective value $\hat{g}_{N'}(\bar{x})$ using the following formula:

$$\hat{g}_{|N'|}(\bar{x}) = \sum_{s \in S} \sum_{e \in E} C_s \cdot x_{e,s} + \frac{1}{|N'|} \sum_{n \in N'} \cdot \left(\sum_{p \in P} \sum_{e \in E} (o^n_{p,e} \cdot O + u^n_{p,e} \cdot U) + \sum_{p \in P} \sum_{e_1 \in E} \sum_{e_2 \in E} W_{e_1,e_2} \cdot v^n_{p,e_1,e_2} \right) \tag{11}$$

The optimality gap can now be estimated by $\hat{g}_{|N'|}(\bar{x}) - \bar{v}^{|M|}_{|N|}$ The number of scenarios is then iteratively increased to identify the point at which the optimality gap is small enough to be accurately used.

4.3 Robust optimization

To ensure that the model is robust, we introduce parameter Γ, which indicates the level of understaffing (relative to demand) beyond which a scenario is flagged as a 'bad scenario'. This is represented by the binary variable y^ω. Additionally, we define parameter $\mathcal{E}$, which specifies the maximum allowable ratio of 'bad scenarios' in a single run. This approach ensures that the understaffing boundary is respected while allowing for the exclusion of certain scenarios—these may arise due to extreme sample outcomes in the random selection of the demand parameter.

Table 8 summarizes the new notation. Constraint 12a ensures that the binary variable y^ω is set to 1 if a scenario exceeds the understaffing boundary. Constraint 12b limits the total number of 'bad scenarios' to comply with the scenario boundary C.

Parameter	Definition
Γ	Understaffing boundary, indicating the ratio of understaffing allowed on one day.
$\mathcal{E}$	Scenario boundary, indicating the ratio of scenarios that can be flagged as 'bad scenario'.
M	A large number.
Variables	**Definition**
$y^\omega \in \{0,1\}$	Binary variable indicating whether scenario ω violates the understaffing boundary.

Table 8: Notation for robustness assurance

$$\sum_{e\in E}\sum_{p\in P}\left(u^\omega_{p,e} - \Gamma \cdot D^\omega_{p,e}\right) \leq y^w \cdot M \quad \forall \omega \in \Omega \tag{12a}$$

$$\sum_{\omega\in\Omega} y^w \leq \mathcal{E} \cdot |\Omega| \tag{12b}$$

4.4 Delayed break scheduling

We introduce delayed break scheduling to enhance the flexibility of the scheduling model further, thus improving the quality. This approach allows for dynamically adjusting the break times of employees in response to real-time fluctuations in demand (i.e., adjusting the break time for each scenario). We assume this break scheduling after demand has realized to be reasonable as planners currently assign breaks in real time as well, where almost all delay information is already known. To model these changes, the indicator $A_{s,p}$ whether shift s is active during period p is changed to not incorporate breaks, but a second indicator $P_{b,p}$ is introduced that becomes 1 if break b (which is part of a set for each specific shift) is active, and 0 otherwise. We furthermore introduce another second stage decision variable $i^\omega_{e,s,b}$ that states how many employees from pool e of shift s are on break b in scenario ω. Consequently, a second stage variable $y^\omega_{p,e}$ is added to hold the information on how many employees from pool e during period p in scenario ω are currently on a break. The formulation changes are given in Table 9. These changes are not present in any literature as far as we know, and thus newly introduced.

Sets	Definition
$b \in B$	Set of allowable breaks for each shift
Parameter	**Definition**
$F_{s,b,p} \in \{0,1\}$	Indicator whether break b is active in shift s in period p
Second-Stage Variables	**Definition**
$i^\omega_{e,s,b} \in \mathbb{Z}^+$	Number of workers from employee pool e of shift s that are on break b in scenario ω.
$y^\omega_{p,e} \in \mathbb{Z}^+$	number of people from employee pool e on break during period p in scenario ω

Table 9: Notation for delayed break scheduling.

$$w_{p,e} - y^{\omega}_{p,e} - \left(o^{\omega}_{p,e} + \sum_{e' \in E} v^{\omega}_{p,e,e'} \right) + \left(u^{\omega}_{p,e} + \sum_{e' \in E} v^{\omega}_{p,e',e} \right) = d^{\omega}_{p,e} \quad \forall p \in P, e \in E, \omega \in \Omega \tag{13a}$$

$$y^{\omega}_{p,e} = \sum_{s \in S} \sum_{b \in B_s} F_{s,b,p} \cdot i^{\omega}_{e,s,b} \quad \forall \omega \in \Omega, e \in E, p \in P \tag{13b}$$

$$\sum_{b \in B} i^{\omega}_{e,s,b} = x_{e,s} \quad \forall \omega \in \Omega, e \in E, s \in S \tag{13c}$$

$$\sum_{e' \in E} v^{\omega}_{p,e,e'} \leq w_{p,e} - y^{\omega}_{p,e} \quad \forall e \in E, p \in P, \omega \in \Omega \tag{13d}$$

First, the demand satisfaction constraint (Equation 8a) is changed to deduct employees on break, as this is now not incorporated in variable $w_{p,e}$ anymore (new formulation is given in Equation 13a). Secondly, Equation 13b is introduced to calculate the number of employees that are on a break. We introduce Equation 13c to make sure that all employees receive exactly one break. Lastly, we change Equation 8c such that we do not exchange more employees than are actually working, and are not on a break (Equation 13d).

4.5 Summary & conclusion

In this chapter, we answered the research question: *"Which methods are suitable to improve the current shift scheduling approach?"*. For this, we designed three models in addition to the existing scheduling algorithm TAM. First, the DAM takes a different approach, accumulating demand per period, and satisfying this total demand for each period. This model formulation then ignores each separate task, reducing the size of the model. We compared the two models and reduced the number of constraints by around 99.7% (3,213,945 to 8,721) and reduced the number of variables by 99.9% (2,082,481 to 2,025).

In the DAM, there is a risk of a non-accurate translation from separate tasks to total demand. Therefore, we designed a variation on the TAM, that maximizes the tasks assigned from a given schedule. With this model, the DAM can be evaluated on quality (i.e., how bad is the non-accurate translation from tasks to demand).

We furthermore extend the DAM to incorporate stochasticity (SDAM). For this, demand scenarios are generated using the fitted noncentral t-distribution and the model optimizes for each scenario combined. Lastly, we propose two extensions to the SDAM, a robust optimization model that incorporates certainty parameters, which can build in security for KLM while scheduling staff, and a delayed break model, allowing dynamically assigning breaks to employees for different scenarios on the DoO. In Chapter 5, we experiment with the four proposed models to assess the quality, advantages, and disadvantages.

5 Experiments

In this chapter, we experiment with the proposed models of Chapter 4 to identify the performance of the models under varying circumstances. In this chapter, we answer the research question:

"How do the different solution approaches perform in different experimental settings?"

For this, different experiments are designed. In Section 5.1 we evaluate whether the newly proposed DAM assumptions have a significant impact on the model outcome, subsequently, we assess the performance of the model considering runtime. We do this by disallowing understaffing, which is also the case in the TAM. We determine the optimal settings for running a stochastic model and then compare the behaviour of the SDAM with the DAM in Section 5.2. We do this by comparing the results using well-known metrics such as the Value of the Stochastic Solution (VSS) and the Expected Value of Perfect Information (EVPI). Lastly, we perform a sensitivity analysis on different model settings to examine the performance of the model in those settings in Section 5.3. All models are tested on 7 datasets, corresponding to one week without abnormal disruptions, the datasets can be found in Appendix C.

Experimental settings During all experiments, we use the settings as displayed in Table 10, unless specifically noted in the section. As sets, we will show in Subsection 5.2.1 that forty scenarios result in an accurate estimation. Furthermore, we define a test set as 'very large', in this case 750 scenarios. We use 10 replications to balance computational efficiency and the accuracy of our results, ensuring that the observed performance reflects general trends rather than random fluctuations. The size of the period set is 222 as we schedule from 05:30-23:59 with a period length of 5 minutes. Furthermore, we include 2 employee pools: D1_TM representing the team members of department D1, and D2_TM representing the team members of department D2. We note that due to the faster solution approach, the preprocessing step to reduce the number possible shift types chosen can be neglected. Therefore S increases from 50 to 105. We decide not to increase the number of break possibilities per shift, as 5 already provides a lot of flexibility to the model.

As cost parameters, we determine the shift costs to be [redacted], resulting in 200 (numbers are multiplied by a random factor) euros per shift. We determine the understaffing cost to be 975, as it is assumed [redacted] We do not penalize overstaffing as it is already incorporated in shift costs. We set the costs of exchanging an employee to another pool to one, as KLM wants to minimize this, but always prefers to exchange rather than have understaffing. We set the maximum number of exchanges to 100 during the day, which, based on performed experiments, we never expect to exceed. Lastly, we value each generated scenario similarly, so the probability of each scenario is set to $\frac{1}{|N|}$.

Sets	Settings	Parameter	Settings
$\lvert N\rvert$	40	$C_{e,s}$	260
$\lvert N'\rvert$	750	U	975
$\lvert M\rvert$	10	O	0
$\lvert P\rvert$	222	$W_{e,e'}$	1
$\lvert E\rvert$	2	$V_{e,e'}$	100
$\lvert S\rvert$	105	π^n	$\frac{1}{\lvert N\rvert}$

Table 10: Default experimental settings

5.1 Demand model quality

To test the model on quality, we designed three different tests, first of all, the impact of accumulating the demand across periods is assessed by running the model proposed in Section 4.1.4 and calculating the ratio of tasks that cannot be assigned to employees. This is presented in Subsection 5.1.1. Secondly, we vary the period length to see the impact of the data transformation complications (as presented in Section 4.1.2), and the results are presented in Subsection 5.1.2. Lastly, we compare the computational performance of the TAM and DAM are compared on runtime for increasing task sets (Subsection 5.1.3).

5.1.1 Number of tasks assigned

To effectively assess the new model formulation, we change model parameter $U = 10,000$ and set $V = |T|$. These settings allow all tasks to be exchanged with minimal costs incurred. Additionally, understaffing is prohibited by choosing a large cost penalty, ensuring alignment with the TAM. Furthermore, a period length of 15 minutes is chosen. Table 11 shows the scheduling results of one week, where columns two and three represent the objective values of the TAM and DAM respectively. The last column shows the percentage of tasks that can be assigned using the evaluation model, and thus the quality of the DAM compared to the TAM. Figure 12 serves as an example of the output of a model, where the blue and orange bars represent the demand per period for both employee pool D1_TM and D2_TM. Furthermore, the black line is the staffing at each period.

Scheduled day	Obj Value TAM	Obj value DAM	Number of tasks assigned
2024-10-21	2101	2047	99.5%
2024-10-22	1940	1850	99.2%
2024-10-23	1910	1842	99.2%
2024-10-24	2010	1900	98.9%
2024-10-25	2205	2083	98.6%
2024-10-26	2304	2139	98.0%
2024-10-27	2201	2121	99.0%

Table 11: Percentage of tasks that can be assigned with the demand accumulation model

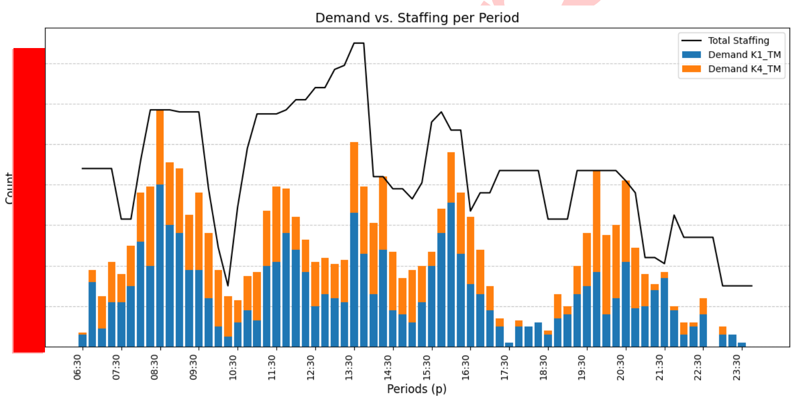


Figure 12: Staffing output of model 2024-10-21

The table shows that the objective value of the DAM is lower, indicating that fewer employees must cover all periods. This outcome is expected, as the model is a slightly relaxed version of the TAM, as explained in Section 4.1.2. In Figure 12, all demand is covered in the DAM. However, to evaluate the extent of the relaxation made, we examine the last column of Table 11, which reveals that most tasks can be assigned to employees. However, a small number of tasks remain unassigned and would require scheduling additional employees to be completed. We expect this margin to be minimal, posing no significant issue for KLM. We do note that the issues result from the transformation from tasks to demand, in which the period size chosen matters significantly, therefore, Section 5.1.2 explores the 'best' period length.

5.1.2 Variable period length

In this section, we discover how a varying period length affects the split of tasks explained in Section 4.1.2, and thus how realistically the DAM represents the reality. We test the model with a period length that varies: $[1, 2, .., 30]$, it is assumed that breaks span an entire period, therefore, the length of the break varies per chosen period length, but lies as closely around the 45 minutes (which is used by KLM). Using larger periods, like 18 minutes, results in a slightly deviating break time (i.e., 2 periods, thus 36 minutes).

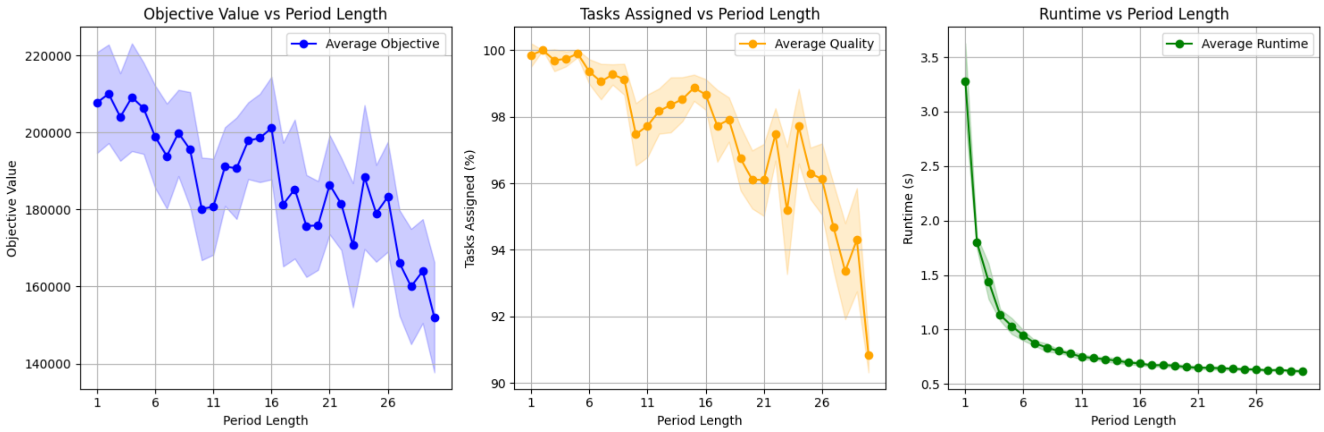


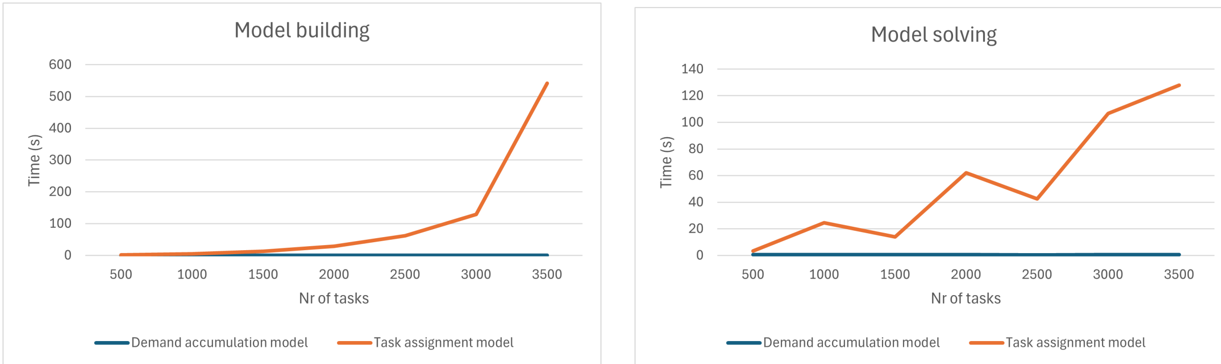
Figure 13: Experimenting with different lengths of periods

Figure 13 presents three plots. The first plot shows the objective value of the DAM (left), while the second plot displays the percentage of tasks assigned with the resulting shifts, as calculated by the evaluation model (middle). It is evident that as the period length increases, the number of tasks that can be assigned (as retrieved from the evaluation model) decreases. This decline is also observed in the objective value. The reason for this is that when larger periods are selected, initially smaller tasks, and eventually even larger ones, no longer span half of the period and thus are excluded from the input set. Consequently, the DAM optimizes for a situation that does not reflect reality (and thus results in a lower objective value), leading to the evaluation model's inability to assign all tasks in such cases. This indicates that using a more granular demand representation (and thus representing the moments the task occurs more accurately) is more valuable. In Subsection 4.1.2, we present a hypothesis regarding task splits between shifts when periods are small. However, we find this issue to be minimal, as nearly 100% of tasks can be assigned in all periods from 1 to 5. This is due to the significant variation in shift types, which allows the model to be flexible in assigning tasks to shifts that are not about to end. The problem could have been more present when only two shift types and task that spans exactly between these shift types.

Better quality solutions at shorter periods indicate that if runtimes allow, the period length chosen should be small. We see that the percentage of tasks assigned is extremely high until the period length becomes larger than 9, runtime declines exponentially when shortening the period length. Thus we decide that a period length of 5 minutes gives an accurate presentation of the demand, but does come with a smaller runtime.

5.1.3 Runtime comparison

To investigate the runtime of TAM and DAM, we use the parameters as presented in the number of tasks assigned. At KLM, around 3400 tasks are scheduled each day. Therefore, we compare the scalability of both models up to 3500 tasks. The time to build (create the sets, parameters, variables, and construct constraints) and solve the model is presented in Figure 14.



(a) Model building time (b) Model solving time

Figure 14: Runtime comparison

A distinct advantage of the DAM becomes evident when comparing its computational performance to the TAM. The TAM experiences a building time increase of up to 9 minutes, with solving times exceeding 2 minutes as the number of tasks grows. In contrast, the DAM maintains a consistent building time of approximately 0.35 seconds, regardless of task count. Similarly, its solving time remains steady at around 0.44 seconds.

This difference can be explained by analyzing the implications of increasing task numbers on both models. For the TAM, a higher number of tasks results in more overlapping tasks (cliques), leading to more sets and constraints. Additionally, constructing sets of shifts capable of performing tasks becomes more time-intensive, and each task requires a binary variable to indicate its assignment to a shift. This increasing number of sets, variables, and constraints causes both building and solving time to increase.

On the other hand, the DAM avoids these complexities. The number of sets remains constant since the periods are fixed, and only the demand within those periods changes. Moreover, the number of variables stays unchanged, representing integer values corresponding to demand and scheduled workers. While these integer values increase with higher demand and larger schedules, this does not significantly impact the model's building or solving performance, resulting in consistent efficiency across different task sets.

5.2 Deterministic vs stochastic approach

In this section, we move from a deterministic approach to a stochastic one. This means that we do not just transform the task set given to us to demand per period, but also generate a set of scenarios, of which the start time (and thus end time) is delayed or taken forward by sampling from the Noncentral t-distribution as presented in Section 2.2. We then optimize for the different scenarios simultaneously using the SDAM, Figure 15 shows the demand per period when we take the expected demand and 16 shows two possible demand scenarios, where reffig: Delay1 and 16b are two (randomly selected) generated demand scenarios.

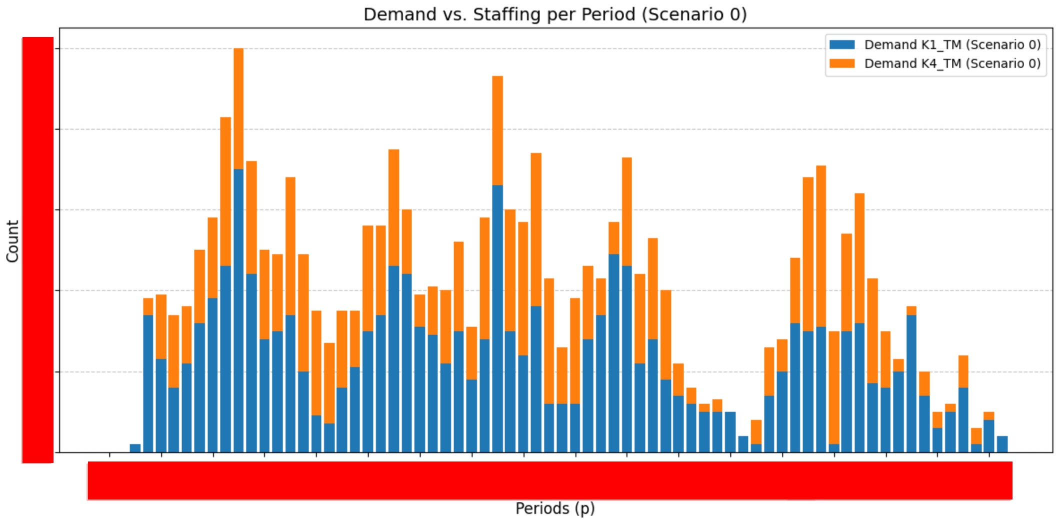
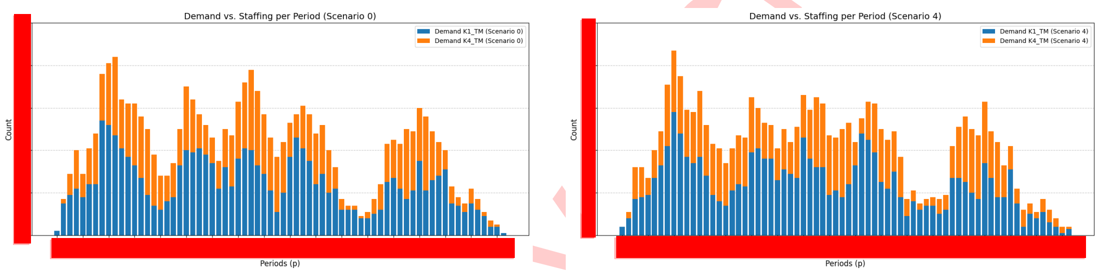


Figure 15: Expected demand



(a) Stochastic scenario set 1

(b) Stochastic scenario set 2

Figure 16: Experimenting with Shift vs Understaffing costs

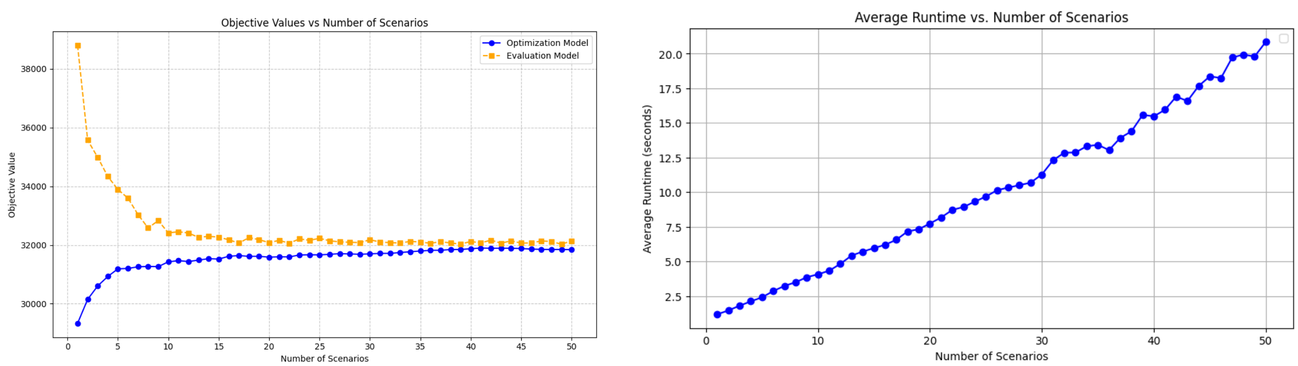
As we can see, the deterministic scenario shows more variation in the demand for each period, with peaks around 100 and a few around 100. The two (randomly sampled) scenario sets show more of a flattened structure, which also represents reality to some extent as task execution does not follow a strictly deterministic pattern. This is to be expected since tasks can get delayed, and thus the peaks spread out, with the highest peaks around 85. Additionally, because the demand is sampled in an uncorrelated manner, the peak flattens a bit more than in reality. Because of this, a stochastic model can be expected to yield solutions where fewer employees need to be scheduled.

So far, the parameters have been chosen such that understaffing does not occur, reflecting the behaviour of the TAM (by setting the costs for understaffing sufficiently high to ensure the acquisition of additional shifts). However, when transitioning to the SDAM, these parameters lead to significant overstaffing. This arises because, for each scenario, the model avoids any understaffing by acquiring a large number of employees for each period as the scenario sets grow. In Subsection 5.2.1, we determine the right settings for the SAA. In Subsection 5.3.1, we analyze the impact of different cost parameters. We exclude experiments involving changing the costs of exchanges since it is reasonable to assume that KLM will always exchange a task if employees are available. Setting this cost to zero, however, would lead the model to suggest unnecessary task exchanges, therefore, a cost of one is kept throughout all experiments. Seven days of scheduling are run, after which the results are averaged and presented. Subsections 5.2.2 and 5.2.3 display the metrics used to compare the DAM and SDAM.

5.2.1 Sample Average Approximation

We perform the SAA to determine the right number of scenarios. For this, we iteratively increase the sample size $|N|$ and each time optimize $|M| = 10$ replications, in Figure 17a this is represented as the blue line and is a lower bound as represented by Equation 10. From these M replications, the average solution ($\bar{x}$) is used to evaluate the solution on a much larger set $|N'| = 750$ as mentioned in Equation 11. The objective value of this evaluation is given by the orange line.

We first generate an evaluation set of 750 scenarios, which remains fixed across all iterations. Initially, we start with a single sampled scenario. In each subsequent iteration, we add one additional scenario to the existing set, rather than resampling. As a result, every new scenario set contains all previously sampled scenarios. This approach reduces variation in the optimization model's objective value but introduces the risk of bias. For example, if the first sampled scenario is particularly unfavourable, its influence persists throughout all subsequent tests. Therefore we need to perform this experiment multiple times to see if the gap between the optimization and evaluation model is indeed small. Figure 17a shows how the objective values behave over an increasing scenario set, while Figure 17b shows the runtime of the algorithm.



(a) Objective value with increasing number of scenarios $|N|$

(b) Runtime of algorithm with increasing number of scenarios $|N|$

Figure 17

The figure shows convergent behaviour towards the 'true' objective value by running the evaluation model. The optimization model approaches this objective value quite fast, since runtime increases linearly and does not restrict our model into limiting the number of scenarios to a small set, we choose 40 scenarios as at this point, the objective value of both models is very close. In the remainder of this research, all experiments are performed with $|N| = 40$ and $|M| = 10$.

5.2.2 Value of the Stochastic Solution

Directly comparing costs of the current situation TAM and the newly proposed SDAM is inaccurate as the TAM covers tasks and disallows understaffing. In Section 5.1, we prove that the DAM can be used to represent the current situation properly, therefore, we evaluate the performance of the SDAM against the DAM using the VSS. The VSS quantifies the benefit of solving the problem stochastically rather than relying on a deterministic optimization based on expected demand.

To compute this, we first solve the SDAM (Equation 7) on a scenario set of size $|N|$, obtaining the optimal stochastic objective value f^s. Next, we solve the DAM (Equation 3) using expected demand values from the distribution, yielding solution x^{ED}. This solution is then evaluated across all scenarios $n \in N$, resulting in the expected deterministic objective value f^{ED}:

$$\min f^{ED} = \sum_{s \in S} \sum_{e \in E} C_{e,s} \cdot x^{ED}_{e,s} + \sum_{n \in N} \pi^n \cdot \left(\sum_{p \in P} \sum_{e \in E} \left(o^n_{p,e} \cdot O + u^n_{p,e} \cdot U \right) + \sum_{p \in P} \sum_{e \in E} \sum_{e' \in E} W_{e,e'} \cdot v^n_{p,e,e'} \right). \tag{14}$$

The VSS as percentage of the solution for the expected demand is then calculated as:

$$VSS = \frac{f^{ED} - f^s}{f^{ED}}. \tag{15}$$

Figure 18 illustrates the results across seven datasets, where boxplots represent the possible savings per task set. The results indicate potential savings ranging from 4.6% to 36.5%, demonstrating the advantages of stochastic optimization. These findings highlight that relying on expected demand alone does not accurately reflect real-world variability, often leading to suboptimal schedules. In contrast, the SDAM provides more robust solutions that adapt better to different scenarios.

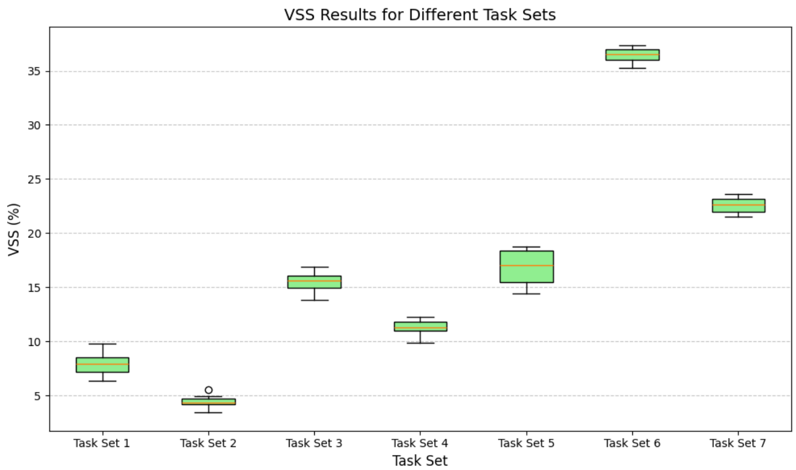


Figure 18: Value of the Stochastic Solution

5.2.3 Expected Value of Perfect Information

The EVPI quantifies the value of knowing the exact task schedule on the DoO, indicating how much effort should be invested in accurate estimations. To compute this, we first sample a scenario set of size $|N|$ and optimize the DAM for each scenario $n \in N$, obtaining an objective value f_n. The expected deterministic solution value is then given by:

$$\mathbb{E}[f^{\text{det}}] = \frac{1}{|N|} \sum_{n \in N} f_n. \tag{16}$$

Next, we solve the SDAM on the same scenario set, yielding an expected stochastic objective value f^s. The EVPI as percentage of the solution for the stochastic model is then computed as:

$$\text{EVPI} = \frac{f^s - \mathbb{E}[f^{\text{det}}]}{f^s} \tag{17}$$

Figure 19 illustrates the results, suggesting a potential cost reduction of approximately 7% per task set. This implies that if KLM had perfect task schedule knowledge, a 7% cost reduction could be achieved by optimizing a single deterministic scenario rather than using a stochastic approach. Notably, these results are based on a six-month planning horizon, where getting an accurate estimation is particularly challenging. Furthermore, when scheduling with perfect information about task details only the 7% of cost reductions could be achieved. This highlights that the SDAM already captures the majority of potential savings, significantly improving scheduling efficiency while leaving only a smaller margin for further optimization.

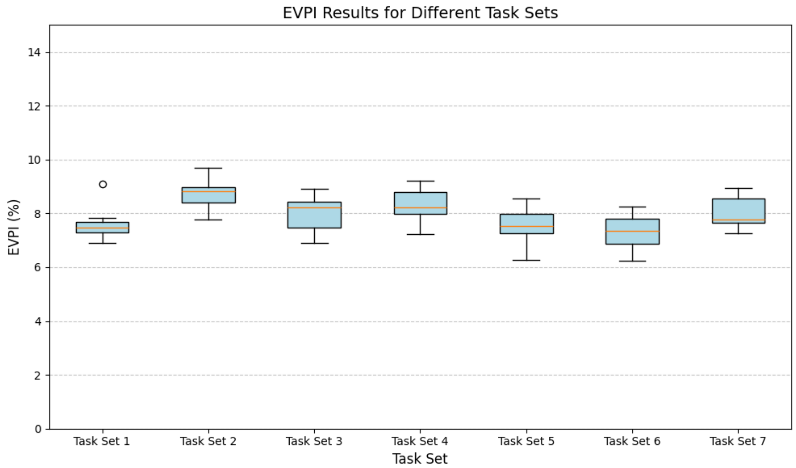


Figure 19: Expected Value of Perfect Information

5.3 Sensitivity analysis

In this section, we will perform a sensitivity analysis on the input parameters (Subsection 5.3.1). Furthermore we will experiment with an increasing correlation coefficient of task delays in Subsection 5.3.2. In Subsection 5.3.3, we compare explicitly modelling robustness vs the stochastic approach. We furthermore experiment with different exchangeability settings in Subsection 5.3.4. Lastly, we look at the benefits of delaying the break scheduling until the DoO in Subsection 5.3.5.

5.3.1 Cost parameters: Understaffing vs Shift costs

We consider 15 different values for C, ranging from 200 to 900, while U varies between 500 and 900. Figure 20 shows how the objective value changes across different parameter settings.

The shift cost parameter primarily influences the objective value, while the impact of the understaffing parameter is less pronounced. We mentioned in Section 4.2 that the model rather schedules more shifts than allowing understaffing to occur. This experiment reinforces that statement, as we see that the understaffing parameter changes affect the objective value much less than the shift costs. We can see an (almost) linear increase in objective value with regard to the shift costs, the higher these costs get, we slowly start to see the curve flatten out.

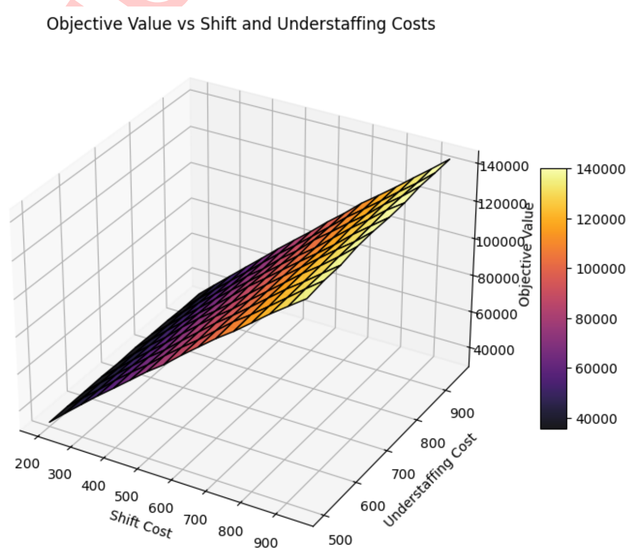


Figure 20: Objective value across different parameter settings

5.3.2 Correlated vs. Uncorrelated Stochastic Behavior

While identifying whether task delays are correlated or uncorrelated is beyond the scope of this thesis, it is expected that external factors like weather or IT failures can cause simultaneous delays across multiple tasks. These correlations can significantly impact scheduling costs. To evaluate this effect, we generate demand scenarios with varying correlation levels using a multivariate normal copula, which preserves the marginal delay distributions while introducing structured dependencies (Meyer, 2009). The correlation matrix defines the strength of dependency between tasks, and delays are sampled by transforming copula-generated values into task-specific distributions. This ensures that tasks experience similar disruptions when correlation is high. By systematically increasing the correlation coefficient, we analyze its impact on scheduling performance. Figure 21 illustrates how objective values evolve as correlation increases.

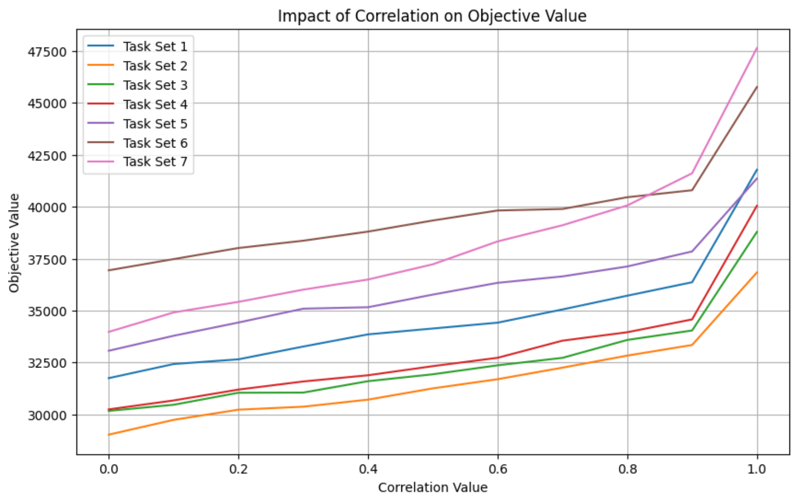


Figure 21: Objective value for different correlation coefficients

Higher correlation values lead to more synchronized delays across tasks, increasing the likelihood of bottlenecks in the schedule. For example, a correlation coefficient of 1 means that all demands shifts the exact same number of periods, this results in multiple scenarios with high peaks at different periods in time. When the correlation coefficient is lower, more peak flattening occurs and some tasks may run late while others remain on schedule, presenting a more favourable picture when scheduling. Therefore, highly correlated delays result in increased staffing requirements. This results in a higher objective value as the shifts must be chosen to cover worst-case scenarios more frequently.

In real-world operations, correlation is typically not uniform across the entire task set but exhibits temporal patterns throughout the day. Early in the day, delays often propagate across tasks due to initial disruptions, constrained resources, or interdependencies in the workflow. As the day progresses, some of these delays may dissipate, while others persist due to cumulative effects. By the afternoon, correlation structures may differ due to operational adjustments, resource reallocation, or shifting demand patterns. For this, other methods (such as autoregressive approaches) can be used as the approach allows for a more realistic representation of delay propagation, ensuring that correlation structures evolve dynamically rather than being imposed uniformly. Future research could explore the calibration of such models using historical delay data to improve the accuracy of scenario generation in stochastic optimization.

5.3.3 Robust Optimization

In this section, we shift from a stochastic optimization model to a robust optimization approach. As understaffing costs arise from various reasons, this is hard to estimate. Therefore, our parameter estimation is hard, to get rid of this parameter, we apply a robust optimization. This transition's primary goal is to minimise shifts' costs while ensuring a certain percentage of demand coverage, as outlined in Section 4.3. This is done by introducing two safeguard mechanisms to handle uncertainty in demand while balancing operational feasibility.

First of all, under the robust optimization framework, we ensure that understaffing does not exceed a predefined threshold relative to demand. Specifically, the amount of understaffing is constrained to a maximum of $\Gamma \cdot D^{\omega}_{p,e}$, where Γ represents the understaffing boundary, and $D^{\omega}_{p,e}$ is the demand for employee pool e during period p in scenario ω. By allowing for a maximum permissible amount of understaffing, we ensure that the schedule remains resilient to demand fluctuations, preventing excessive shortages that could disrupt operations.

In addition to controlling the extent of understaffing, we manage the scenario viability. For this, we introduce a second safeguard to manage the number of scenarios that can be deemed “bad” (i.e., scenarios that exceed the understaffing threshold). We have defined parameter $\mathcal{E}$, which specifies the maximum allowable fraction of scenarios that can be flagged as bad. These two strategies combined ensure that the staffing plan is both resilient and realistic, accounting for potential demand fluctuations while maintaining feasible staffing levels.

Through this robust optimization approach, we minimize the risk of under-resourcing and enhance the flexibility and reliability of the staffing system, ensuring that demand is met under typical and extreme conditions. We set the understaffing parameter to $U = 0$ and run the SDAM model with robustness constraints (Equation 12a and Equation 12b) and solve the model for 10 different scenario/understaffing boundary. Figure 22 shows how the objective value changes if the robustness parameters change. For example, the left bottom of the graph shows what happens if we want to satisfy 90% of demand for 80% of all scenarios, while the right top of the figure displays the objective value if we want to satisfy 100% of demand for 100% of all scenarios.

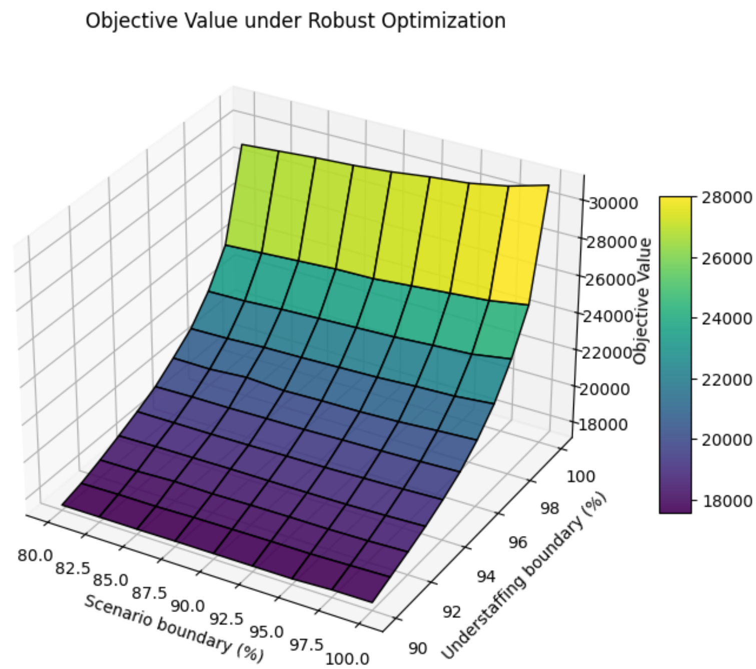


Figure 22: Objective value offset against robustness parameters

The figure shows two key relationships. As expected, costs rise as the boundary parameters increase. However, the relationship between the understaffing parameter and the objective value is much more pronounced compared to the scenario boundary and the objective value. This is because the generated scenarios are highly similar (meeting 80% of scenarios is often very similar to meeting 90%) as many tasks have independent delays that tend to cancel each other out when aggregated at the period level. Due to this minor difference, we focus on the percentage of demand that must be satisfied. Specifically, if 100% of scenarios need to be covered, we determine the number of staff required, as shown in Table 12 (data is scaled such that 100% needs 100 employees). The table illustrates that increasing demand coverage leads to an exponential rise in staffing needs. For example, in this run, only 1.3 additional employees are needed to increase coverage from 90% to 91%, whereas covering 99% to 100% requires 16.8 more employees. It is important to note that the table demonstrates relative rather than absolute increases. The exact number of additional employees required to cover 100% of scenarios depends on the total number of sampled scenarios—if more scenarios are considered, the required staffing level to ensure full coverage will also increase.

Percentage	Staff Needed
90%	58.4
91%	59.7
92%	61.1
93%	62.4
94%	64.4
95%	66.4
96%	68.5
97%	71.8
98%	76.5
99%	83.2
100%	100

Table 12: Staff needed for different percentages

When we compare the Stochastic and robust approach, we run the SDAM on the same dataset as the robust optimization test (2024-10-22). We run 10 replications and present the number of people needed in Figure 23. We see that, on average, we need 94.0 employees, corresponding to between 99-100% certainty to cover all demand. We can conclude from this that even though adding these last percentages of demand coverage and thus a lot of staff remains worth it due to the high costs of delaying a flight.

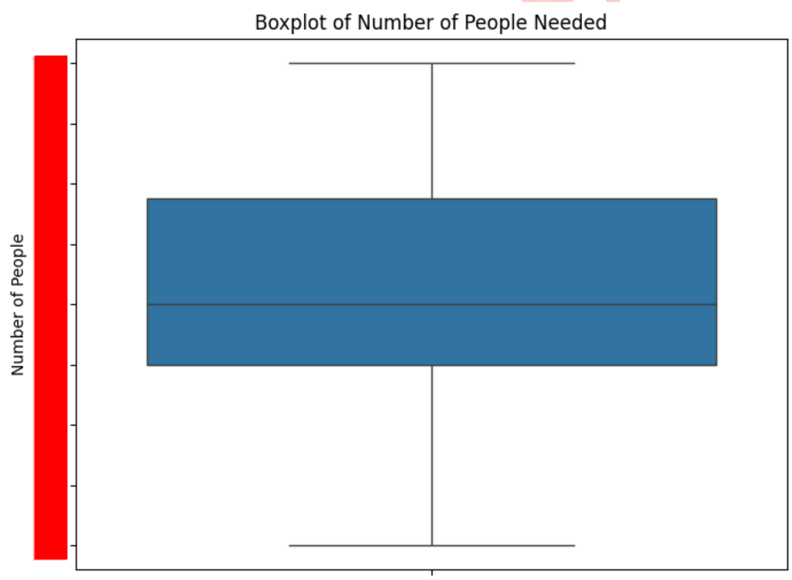
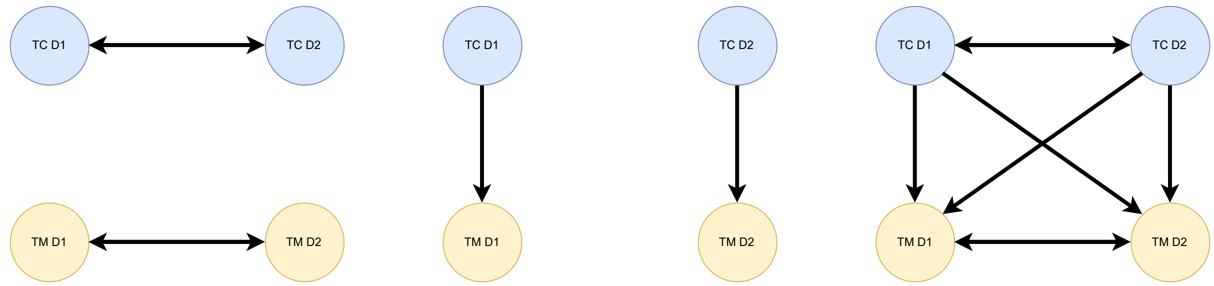


Figure 23: Number of staff needed in SDAM

5.3.4 Exchangeability Experiments

In our experiments, we analyze the impact of allowing employee exchangeability between two departments (D1 and D2) and skill groups (TM and TC). Initially, only Team Members (TMs) were eligible for exchange. To extend our analysis, we now include Team Coordinators (TCs) in the dataset to evaluate the influence of exchangeability more comprehensively. First, we consider a scenario where no exchangeability is allowed, meaning employees remain within their respective departments without any possibility of transfer. Next, we introduce horizontal exchangeability, where departments can exchange employees. However, all tasks must still be performed by employees with the appropriate skill set (Figure 24a). In the third scenario, we examine vertical exchangeability. Here, departments do not exchange staff, but TCs are allowed to perform tasks assigned to TMs. However, TMs cannot perform TC tasks due to certification requirements (Figure 24b) Lastly, we consider cross-exchangeability, where both horizontal and vertical exchanges are permitted. In this case, employees can be exchanged between departments, and TCs can perform tasks of TMs (Figure 24c).This setup allows us to assess the impact of different levels of exchangeability on workforce flexibility and scheduling efficiency.



(a) Horizontal (b) Vertical (c) Cross

Figure 24: Exchange Relations

The set E is expanded to incorporate the TCs of both departments, resulting in $|E| = 4$. For shift costs, [redacted] leading to a total of 280 euros per shift. We generate 10 scenario sets and conduct four tests: one without exchangeability and three based on the models presented in Figure 24. Table 13 presents the average number of shifts required for each employee pool (all numbers are scaled such that the largest number becomes 100), while the last column shows the average objective value. Additionally, Figure 25 visualizes the objective values across all 10 scenario sets/replications.

Test	D1_TM	D2_TM	D1_TC	D2_TC	Total Staff	Objective Value
No Exchangeability	43.0	31.9	14.1	10.9	100.0	59647.62
Vertical Exchangeability	40.3	29.4	15.1	11.9	96.8	57505.94
Horizontal Exchangeability	40.7	30.1	12.9	10.0	93.7	54895.88
Cross Exchangeability	38.6	28.5	13.6	10.6	91.3	53589.89

Table 13: Comparison of Exchangeability Scenarios for D1 and D2

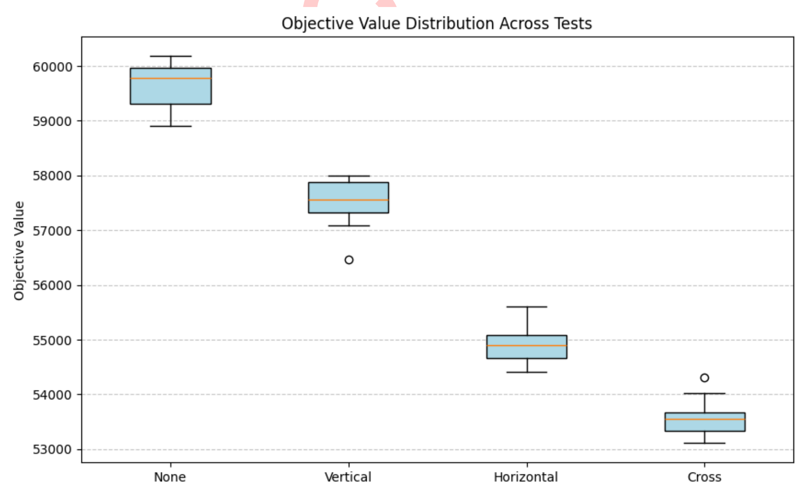


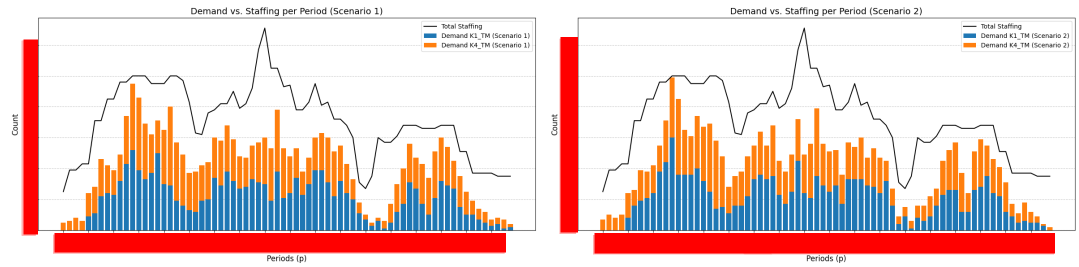
Figure 25: Objective value offset against different exchangeability relations

Figure 25 demonstrates that allowing exchangeability significantly affects the number of employees required. Vertical exchangeability has the smallest impact, resulting in a cost reduction of 3.6% (compared to no exchangeability). Horizontal exchangeability leads to a greater improvement, with an 8.0% reduction in the objective value, compared to the 3.6% improvement seen with vertical exchangeability alone. As expected, allowing both horizontal and vertical exchanges (cross-exchangeability) results in the highest cost reduction, reaching 10.2%. Additionally, Table 13 reveals that when only vertical exchangeability is allowed, it is advantageous to hire more TC employees. This is because TCs provide greater flexibility for planners, which helps optimize staffing.

5.3.5 Delaying Break Scheduling

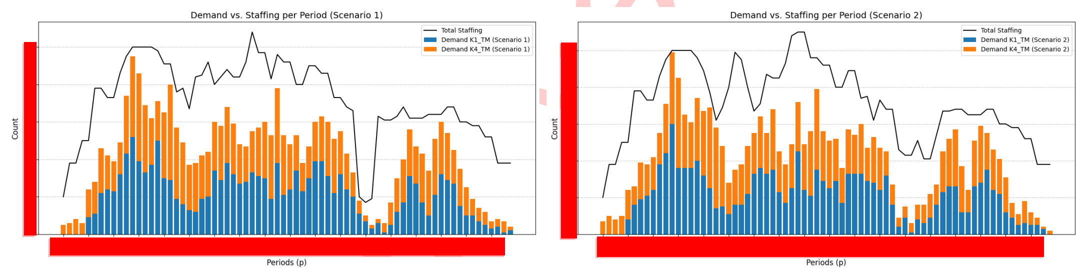
To assess the impact of delaying break scheduling to the DoO, we compare the SDAM with the delayed break model, as presented in Section 4.4. Unlike the SDAM, where shifts and breaks are predefined as a combined set, the delayed break model separates them. Consequently, Set S is modified: its size is reduced from $|S| = 105$ to $|S| = 21$, now representing the shifts only. A new Set B is introduced, with $|B| = 5$. Additionally, new decision variables and constraints are introduced (which are also described in Section 4.4).

To evaluate performance, we generate a scenario set and simultaneously solve both the SDAM and the delayed break model on the same dataset. Figure 26 shows the demand and staffing of scenarios 1 and 2 (randomly chosen) of the SDAM whilst Figure 27 shows the same for the delayed break model. The differences in objective values (SDAM - delayed break) are presented in Figure 28.



(a) Scenario 1 (b) Scenario 2

Figure 26: SDAM staffing vs demand



(a) Scenario 1 (b) Scenario 2

Figure 27: Delayed break staffing vs demand

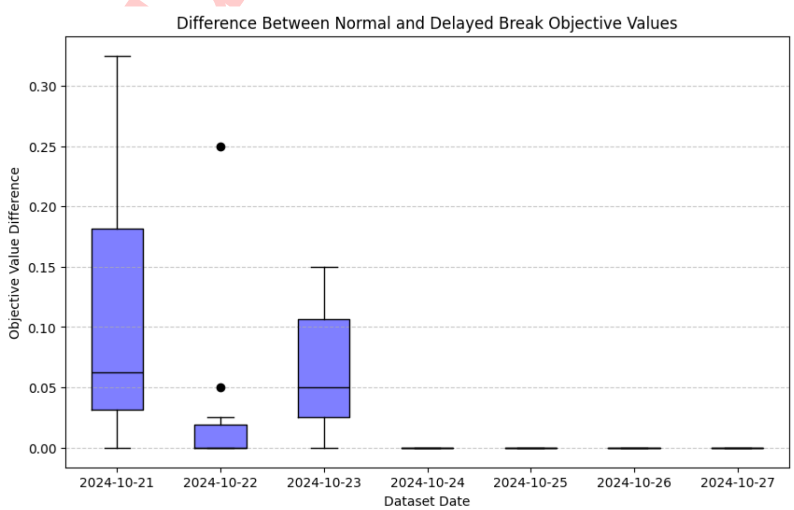


Figure 28: Objective value differences

Figure 27 shows that the staffing is different for the two scenarios presented, this is due to the difference in break allocations. In Figure 26, we can see that staffing is equal over all periods. We observe that, across all datasets the benefit of delaying the break allocation, is negligible as a maximum of 0.3 savings on objective value (corresponding to a cost reduction of 0.001%. This can be explained by looking at KLM's conservative approach: since understaffing incurs a higher cost than hiring additional employees, (almost) every period experiences high overstaffing. As a result, the SDAM can allocate breaks without causing understaffing in any of the 40 scenarios already, and does not need a delayed break allocation to manage this. Only in a few replications did we observe cases where scheduling the break with the shifts resulted in understaffing, leading to a lower objective value when the delayed break model is used.

We believe that the value of the delayed break model can become bigger in two cases. First of all, the uncorrelated task delays in the scenarios cause all scenarios to be fairly similar, which allows the SDAM to schedule one break that fits all. We believe that if task stochasticity is correlated, the delayed break model does yield better solutions than the SDAM. Furthermore, if less conservative parameters are used, for example in a different scheduling problem, the impact of delayed break scheduling is likely to be more pronounced. In such cases, the model would have fewer periods with large overstaffing, giving the model fewer opportunities to assign a single break that accommodates all scenarios.

The computational performance of both models is slightly different as more variables and constraints get introduced in the delayed break model, increasing the complexity of this model. Over all datasets and replications, the average solving time of the delayed break model is 9.6 seconds, while the SDAM takes 6.2 seconds. With the current settings and sizes of the problem, the delayed break scheduling can be preferred over the SDAM, as it is computationally still fast enough and will never be of worse quality than the SDAM.

5.4 Summary & conclusion

In this chapter, we answered the research question: *"How do the different solution approaches perform in different experimental settings?"*. We first compared the currently used TAM with our new DAM to answer this question. In the current scheduling model, no understaffing is allowed, we forced this by applying a high penalty for the understaffing and obtained similar results, with slightly lower objective values. This is due to exchangeabilities between periods instead of full tasks, and some tasks that cannot be assigned. The latter we subsequently addressed by running the task assignment model. It turned out that if periods remain short, we can accurately assess the tasks to be done and obtain a solution where 99-100% of tasks can be assigned. We can conclude that the DAM is the superior model by looking into the runtime, which for real-life instances remains around 0.5 seconds, while the TAM increases with task size up to around 10 minutes.

To apply the stochastic modelling approach (SDAM), we first applied the SAA method to determine the number of scenarios necessary for the optimization model to generate a realistic picture of the DoO. From this, we conclude that 40 scenarios are appropriate. We then compared the DAM with the SDAM and concluded that the stochastic approach consistently outperforms the deterministic approach by consistently lower objective values. The Value of the Stochastic Solution showed that running the stochastic version of the model compared to the deterministic version results in savings of 5 to 36%. Depending on the nature of the input dataset. Furthermore, the Expected Value of Perfect Information shows that obtaining perfect information on the DoO can save around 7% more cost, indicating that we already tackle most of the improvements, and small steps are still to be made.

We performed a sensitivity analysis on different factors that might influence the model. First of all, we constructed a robust optimization model that does not minimize understaffing but satisfies staffing up to the desired levels. This model showed that satisfying a higher percentage of demand and scenarios results in exponentially increasing shift costs. We compared the SDAM with the robust optimization, and from this we concluded that the estimated understaffing costs in the stochastic program are so high that it automatically satisfies a large percentage of demand and scenarios. Therefore, the robust optimization model results in fewer shift costs, but also a larger risk of understaffing.

We performed a test on the correlation between task delays to see the impact on the objective value. From this we can conclude that correlation does have a significant impact, therefore, a proper correlation analysis should be done to create scenarios that do reflect reality better.

We evaluated the impact of exchangeability on workforce costs. Our results show that allowing exchangeability significantly reduces costs by optimizing staff utilization. Vertical exchangeability, where Team Coordinators (TCs) assist Team Members (TMs), leads to a modest cost reduction of 3.6%. Horizontal exchangeability, where employees can be reassigned between departments while maintaining task-specific skills, results in a greater cost reduction of 8.0%. The highest savings, 10.2%, are achieved when both horizontal and vertical exchangeability are allowed.

Finally, we adjusted the model to allow break allocation to be determined after demand fluctuations have been accounted for. Under the current conservative settings, this adjustment does not provide any noticeable benefits. However, we believe that its impact could become more significant if the conservative approach is relaxed and when task delays exhibit correlation, as this would result in less similar scenarios during the optimization.

6 Conclusion and Recommendations

In this chapter, we answer the research question: *"What conclusions can be drawn from the research at KLM?"* First, we summarize our conclusions about the model formulation and its performance in the experiments in Section 6.1. We then continue with a concise and practical set of recommendations to KLM about their staff scheduling method in Section 6.2. Furthermore, we discuss the limitations of our research and subsequently talk about how future research can improve staff scheduling in Section 6.4.

6.1 Conclusions

We started the research with the research question: *"How can KLM optimize its ground staff scheduling to incorporate robustness alongside effectiveness and efficiency?"*. We have solved this problem by creating two variants of a Stochastic Programming model. One with, and one without explicitly modelling robustness parameters. From the literature review, we concluded that Hur et al. (2019a) model breaks within the shift scheduling algorithm, while Wu et al. (2023) applied a periodic approach. However, this periodic approach does not align with KLM's current scheduling problem, which more closely resembles a task assignment problem. We first formulated a deterministic model using a periodic view of demand instead of the task-based model and concluded that this formulation reduces the model size by around 99.8%. Subsequently, we tested the impact of this transformation on the quality. We concluded that if periods are chosen small enough (5 minutes), the quality remains equal, whilst the runtime decreased from around 12 minutes to 1 second.

We then continued to extend this model formulation to incorporate stochasticity in demand. To tackle this, we fitted the noncentral t-distribution to the dataset to estimate plane delays and used this to delay the tasks that appear in our input dataset. We should note that this leads to some peak flattening, whilst in real life, a larger correlation between periods of the day, and delays of tasks appears. In Section 5.3.2, we showed that the correlation has a significant impact on the model outcome, and thus cannot be neglected. To see the benefits of using a stochastic model, we compared solving the model with expected demand against solving the model for forty scenarios using the Value of the Stochastic Solution and can conclude that there is a cost-saving opportunity between 4.6-36.5%. Whilst achieving this, runtime remains below 20 seconds, indicating that we can solve this problem to optimality whilst remaining in a reasonable runtime. We also found that allowing exchangeability between employee pools can have a positive impact on the number of employees needed during the DoO, where full exchangeability can result in 10.2% cost savings compared to no exchangeability.

We introduced two extensions to the SDAM. First, we allowed the scheduling of breaks to be delayed until after the demand scenarios have been realized. However, with the current parameters and input settings, this extension does not provide any benefits. Nonetheless, we believe that under different settings, the delayed break model could yield better results while maintaining a slightly worse but still feasible computational performance compared to the SDAM. Second, we modified the SDAM to allow explicit control over the level of conservatism, creating a robust model. Given that the cost of not meeting demand is currently much higher than the cost of acquiring additional staff, all models ensure sufficient staffing. However, the robust model shows that increasing the percentage of demand that must be satisfied leads to an exponential increase in shift costs. Covering the final few percent of demand requires disproportionately more employees than covering the preceding percentages.

6.2 Recommendations

Based on this research and the conclusions drawn from the experiments. We propose a list of recommendations for KLM. By incorporating these recommendations in the staff scheduling process, we expect to save costs on ground staff, whilst improving the workflow of the tactical planners.

1. Replace the current scheduling process by the Stochastic Demand Accumulation Model.

   By implementing the SDAM, the added benefit is threefold. First of all, it is possible to combine the Shift Type Optimizer and Shift Optimizer. This will broaden the solution space and thus have the potential to find better solutions. Secondly, incorporating the stochastic scenarios leads to more robust schedules and cost reductions compared to deterministic approaches that do not incorporate Day of Operation deviations in the schedule. Lastly, the runtime of the model is significantly decreased to around 20 seconds, making the algorithm more practical to use.

2. Train staff to be able to work in other departments and skills.

   Our analysis shows that allowing staff exchangeability can significantly reduce costs. Each type of exchangeability, Horizontal, Vertical, and Cross, offers distinct benefits and requires specific training. KLM can evaluate whether the potential cost savings justify the investment in staff training.

3. Consider the trade-offs between the SDAM and the robust optimization model.

   While the SDAM properly weighs off understaffing vs shift costs, it results in a very conservative planning approach and thus high personnel costs. By using the robust model, KLM can determine the desired level of certainty and potentially reduce personnel costs by taking slightly more risk.

4. Collect data on the stochastic variations in task delay and duration.

   The current model derives scenarios from (uncorrelated) delays, however, more factors can influence the tasks start time and durations. As only delays are incorporated, a less accurate representation of reality is given, leading to a too heavy peak-flattening effect. KLM can collect data directly from the ground staff performing tasks (for example by measuring task start delays and task durations) so that an accurate set of scenarios can be generated. This data can then be used to further improve the SDAM model. By measuring fluctuations in task duration, the SDAM model can perform better.

5. Train personnel to work with the model based on periodic demand rather than task assignments.

   While the DAM is a good model for creating shift schedules, it provides less information on specific task assignments per shift. Therefore, personnel must be trained to work with periodic demand, which is crucial to realizing the benefits of the new approach.

6.3 Contributions

As found in the performed literature research, Wu et al. (2023) propose a stochastic shift scheduling algorithm, Poullet and Parmentier (2020) extend it by including breaks as part of preprocessing, and Hur et al. (2019b) takes a step further by incorporating flexible break scheduling as the stochastic scenarios play out. In this research, we have combined the model formulations to integrate stochastic shift scheduling with breaks included in preprocessing. We have extended this approach by introducing exchangeability within periods such that other skills can cover demand. We efficiently model this to ensure that both mutual and one-way relations can be incorporated. Lastly, we formulate the model to handle real-life instances, such as those encountered by KLM.

Wu et al. (2023) compare the SAA method with a heuristic alternative and state that, when using 250,000 scenarios, the heuristic becomes competitive with SAA if the same runtime is considered. However, with a smaller number of scenarios, SAA outperforms the heuristic. We observe that from 40 scenarios onward, the objective value closely approximates the ”true” objective, making it sufficient to limit our scenario count to 40. Consequently, SAA remains the preferred method. Furthermore, Hur et al. (2019b) report that flexible break scheduling can mitigate 94% of understaffing. In contrast, our implementation of delayed break scheduling did not yield such significant improvements, showing only minor gains in a few instances, but no gains in the majority of instances. We explain this by KLM's conservative approach and the largely uncorrelated and relatively similar demand scenarios. However, in other shift scheduling contexts, where understaffing may be preferable to hiring additional employees, delaying break scheduling could have a more substantial impact as Hur et al. (2019b) state.

Practically, our model contributes to the scheduling methods of KLM by speeding up the algorithm. This way, we allow the input sets for the model not to be narrowed and thus the optimization to have a larger potential. Furthermore, KLM only optimizes deterministically, we enable the optimization of different scenarios. We introduce employee exchangeability within departments while allowing both mutual and one-way exchanges. This makes combined optimization between different groups possible, and thus a larger potential for cost savings is introduced.

6.4 Limitations and Further research

We first look at the proposed model (DAM) against the current scheduling method in a deterministic setting. Even though the DAM outperforms the TAM significantly when looking at the computational performance, we see that the assumptions about periods and task allocation results in some insurance about fitting all tasks. Even though we can evaluate our model and thus show that in a certain percentage of the time all tasks can fit, users are not 100% sure. This can be problematic as the current schedulers take a preservative approach when scheduling. Adding to this, the TAM assigns tasks to shifts, which the DAM cannot. This means that planners receive less information from the model. We believe that this should not be a problem, as the goal of the schedulers is to construct shifts and not to assign tasks, it may be difficult to ensure user acceptance.

When looking at the SDAM, a large limitation is the current stochastic scenarios that are chosen. We have decided to only incorporate delays of planes as input for our scenario generation. There are a few problems with this approach. First of all, the delay of a plane is not a one-on-one representative of the deviation of tasks. When a plane is delayed because no towback car is available, the luggage handling staff can leave the plane and continue with another task. However, when passenger boarding extends and leads to delays, the staff is unable to leave, as luggage may have to be taken out of the plane. Furthermore, delay behaviour is different for different periods of the day, and different departments. For example, intercontinental flights usually arrive early in the morning, and arrive often earlier than expected, while short-haul flights usually experience late arrivals. Another downside of the scenario calculation is the exclusion of the appearance of new tasks and task duration, as there is no data gathered for this, it is impossible to accurately quantify what the effect on the tasks is, however, employees do agree that these have a significant effect on the schedule on the DoO.

Furthermore, the SDAM schedules shifts per day. As this ignores rosterability (shifts should be similar across the week), the model should be extended to minimize shift deviations in addition to the current objective. Lastly, another model extension is to postpone scheduling the breaks until the demand has played out. Currently, breaks are incorporated in shift construction (pre-processing of the model) but are neglected and rescheduled by the planners on the DoO, therefore, the assumption that breaks can be assigned as the demand scenario unfolds is justified. Hur et al. (2019b) explores this extension, combining our research with the one from Hur et al. (2019b), which will improve the scheduling process even further.

References

Abernathy, W., Baloff, N., Hershey, J., and Wandel, S. (1973). A three-stage manpower planning and scheduling model–a service-sector example. *Operations Research*, 21:693–711.

Ahmed, S. and Shapiro, A. (2002). The sample average approximation method for stochastic programs with integer recourse. submitted for publication. *Science*, 12.

Akkermans, A., Post, G., and Uetz, M. (2021). Solving the shift and break design problem using integer linear programming. *Annals of Operations Research*, 302:341–362.

Aloulou, M. A., Haouari, M., and Zeghal Mansour, F. (2013). A model for enhancing robustness of aircraft and passenger connections. *Transportation Research Part C: Emerging Technologies*, 32:48–60.

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

Bechtold, S. E. and Jacobs, L. W. (1990). Implicit modeling of flexible break assignments in optimal shift scheduling. *Management Science*, 36(11):1339–1351.

Beliën, J. and Demeulemeester, E. (2008). A branch-and-price approach for integrating nurse and surgery scheduling. *European Journal of Operational Research*, 189(3):652–668.

Bentaha, M., Battaïa, O., and Dolgui, A. (2014). A sample average approximation method for disassembly line design under uncertainty. *Computers Operations Research*, 51:111–122.

Burke, E., De Causmaecker, P., van den Bergh, G., and Van Landeghem, H. (2004). The state of the art of nurse rostering. *J. Scheduling*, 7:441–499.

Cappanera, P., Gangi, L. D., Lapucci, M., Pellegrini, G., Roma, M., Schoen, F., and Sortino, A. (2024). Integrated task scheduling and personnel rostering of airports ground staff: A case study. *Expert Systems with Applications*, 238.

Caprara, A., Monaci, M., and Toth, P. (2003). Models and algorithms for a staff scheduling problem. *Mathematical Programming*, 98(1):445–476.

Cavada, J. P., Cortés, C. E., Henríquez, G., and Rey, P. A. (2023). A ground crew shift rostering model for santiago international airport. *Operational Research*, 23.

Cook, A. and Tanner, G. (2015). European airline delay cost reference values. *University of Toronto Press*, pages 5–9.

Dantzig, G., Fulkerson, R., and Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4):393–410.

Di Gaspero, L., Gaertner, J., Kortsarz, G., Musliu, N., Schaerf, A., and Slany, W. (2007). The minimum shift design problem. *Annals of Operations Research*, 155:79–105.

Di Gaspero, L., Gärtner, J., Musliu, N., Schaerf, A., Schafhauser, W., and Slany, W. (2013). Automated shift design and break scheduling. *Automated Scheduling and Planning: From Theory to Practice*, pages 109–127.

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

Gök, Y. S., Padrón, S., Tomasella, M., Guimarans, D., and Ozturk, C. (2023). Constraint-based robust planning and scheduling of airport apron operations through simheuristics. *Annals of Operations Research*, 320(2):795–830.

Gök, Y. S., Tomasella, M., Guimarans, D., and Ozturk, C. (2020). A simheuristic approach for robust scheduling of airport turnaround teams. In *Proceedings - Winter Simulation Conference*, volume 2020-December, pages 1336–1347. Institute of Electrical and Electronics Engineers Inc.

Helber, S. and Henken, K. (2010). Profit-oriented shift scheduling of inbound contact centers with skills-based routing, impatient customers, and retrials. *Operations Research-Spektrum*, 32:109–134.

Hur, Y., Bard, J. F., Frey, M., and Kiermaier, F. (2019a). An investigation of shift and break flexibility with real-time break assignments using a rolling horizon approach. *Flexible Services and Manufacturing Journal*, 31(1):174–211.

Hur, Y., Bard, J. F., Frey, M., and Kiermaier, F. (2019b). A stochastic optimization approach to shift scheduling with breaks adjustments. *Computers and Operations Research*, 107:127–139.

Ingels, J. and Maenhout, B. (2019). Optimised buffer allocation to construct stable personnel shift rosters. *Omega*, 82:102–117.

Juan, A., Keenan, P., Marti, R., McGarraghy, S., Panadero, J., Carroll, P., and Oliva, D. (2021). A review of the role of heuristics in stochastic optimisation: from metaheuristics to learnheuristics. *Annals of Operations Research*, 320.

Kletzander, L. and Musliu, N. (2019). Modelling and solving the minimum shift design problem. In Rousseau, L.-M. and Stergiou, K., editors, *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 391–408, Cham. Springer International Publishing.

Kletzander, L. and Musliu, N. (2024). Hyper-heuristics for personnel scheduling domains. *Artificial Intelligence*, 334:104172.

Kleywegt, A., Shapiro, A., and Homem-De-Mello, T. (2001). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12:479 –.

KLM Royal Dutch Airlines (2023). Annual report 2023.

Leeftink, A. G., Vliegen, I. M. H., and Hans, E. W. (2019). Stochastic integer programming for multi-disciplinary outpatient clinic planning. *Health Care Management Science*, 22(1):53–67.

Liu, C., Chen, Y., Zhang, Y., Wang, H., Luo, Q., and Chen, L. (2024). Reliable and robust scheduling of airport operation resources by simulation optimization feedback and conflict resolution. *Neurocomputing*, 573:127212.

Liu, M., Liang, B., Zhu, M., and Chu, C. (2020). Stochastic check-in employee scheduling problem. *IEEE Access*, PP:1–1.

Lusby, R. M., Range, T. M., and Larsen, J. (2016). A benders decomposition-based matheuristic for the cardinality constrained shift design problem. *European Journal of Operational Research*, 254(2):385–397.

Maenhout, B. and Vanhoucke, M. (2010). A hybrid scatter search heuristic for personalized crew rostering in the airline industry. *European Journal of Operational Research*, 206(1):155–167.

Meyer, C. (2009). The bivariate normal copula. *Communication in Statistics- Theory and Methods*, 42.

Musliu, N., Schaerf, A., and Slany, W. (2004). Local search for shift design. *European Journal of Operational Research*, 153(1):51–64. Timetabling and Rostering.

NOS Nieuws (2024). Klm kondigt reorganisatie aan: 'pijnlijk voor elke klm-collega'. Accessed: 2024-10-08.

Poullet, J. and Parmentier, A. (2020). Shift planning under delay uncertainty at air france: A vehicle-scheduling problem with outsourcing. *Transportation Science*, 54:956–972.

Santos, G. and Robin, M. (2010). Determinants of delays at european airports. *Transportation Research Part B: Methodological*, 44:392–403.

Scheelhaase, J., Braun, M., Maertens, S., and Grimme, W. (2023). Costs for passengers and airlines due to the significant delays and other irregularities at european airports in the 2022 summer season. *Transportation Research Procedia*, 75:96–105. 12th International Conference on Air Transport – INAIR 2023, The Future of Aviation – is the Sky the Limit?

Society for Human Resource Management (2016). Total financial impact of employee absences report. Technical report, Society for Human Resource Management, Alexandria, USA. Accessed: 2023-10-08.

Sun, X., Wandelt, S., and Zhang, A. (2023). Aviation under the covid-19 pandemic: A synopsis from normalcy to chaos and back. *Journal of the Air Transport Research Society*, 1(1):136–151.

Teymouri, A., Sahebi, H., and Pishvaee, M. S. (2023). Airline operational crew-aircraft planning considering revenue management: A robust optimization model under disruption. *International Journal of Industrial Engineering Computations*, 14:381–402.

Thompson, G. M. (1996). A simulated-annealing heuristic for shift scheduling using non-continuously available employees. *Computers Operations Research*, 23(3):275–288.

van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., and De Boeck, L. (2013). Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385.

van Hulst, D., den Hertog, D., and Nuijten, W. (2017). Robust shift generation in workforce planning. *Computational Management Science*, 14:115–134.

Wang, W., Xie, K., Guo, S., Li, W., Xiao, F., and Liang, Z. (2023). A shift-based model to solve the integrated staff rostering and task assignment problem with real-world requirements. *European Journal of Operational Research*.

Wen, X., Sun, X., Sun, Y., and Yue, X. (2021). Airline crew scheduling: Models and algorithms. *Transportation Research Part E: Logistics and Transportation Review*, 149:102304.

Wickert, T. I. (2019). Personnel rostering: models and algorithms for scheduling, rescheduling and ensuring robustness.

Wickert, T. I., Smet, P., and Vanden Berghe, G. (2021). Quantifying and enforcing robustness in staff rostering. *Journal of Scheduling*, 24:347–366.

Wu, C.-L. (2005). Inherent delays and operational reliability of airline schedules. *Journal of Air Transport Management*, 11(4):273–282.

Wu, Z., Xu, G., Chen, Q., and Mao, N. (2023). Two stochastic optimization methods for shift design with uncertain demand. *Omega (United Kingdom)*, 115.

A Luggage handling process

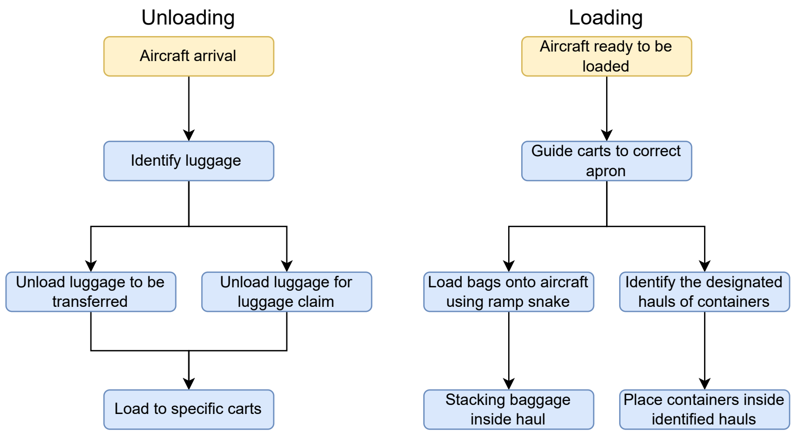


Figure 29: Luggage (un)loading process

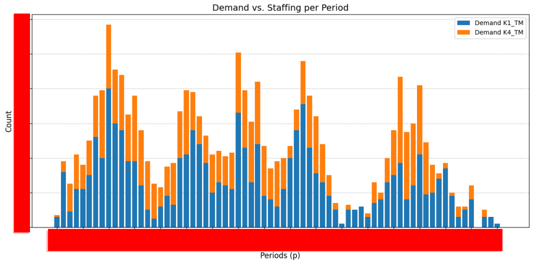
B Maximal cliques explained

[redacted]

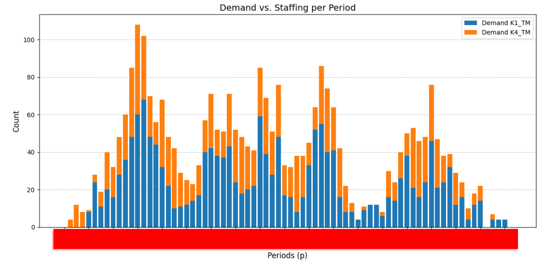
[redacted]

Figure 30: Maximal Clique example

C Datasets used

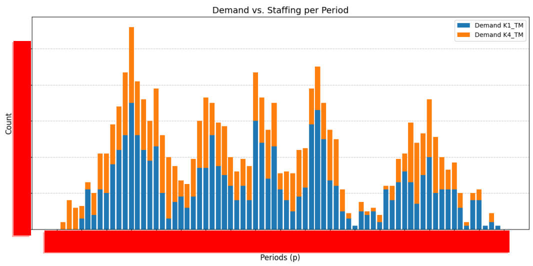


(a) Dataset 21-10-2024

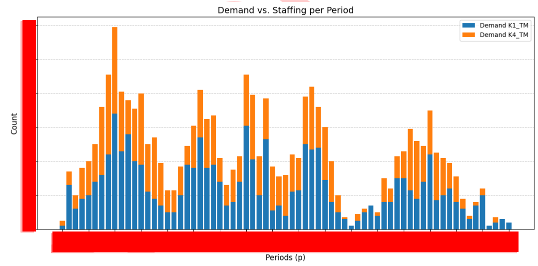


(b) Dataset 22-10-2024

Figure 31

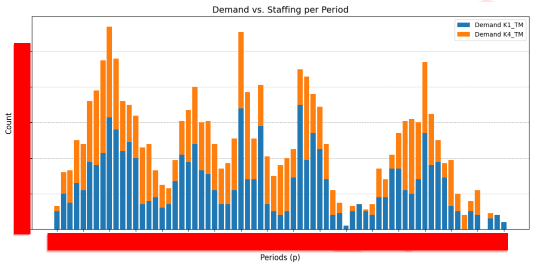


(a) Dataset 23-10-2024

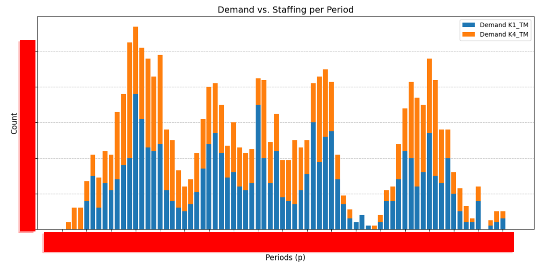


(b) Dataset 24-10-2024

Figure 32



(a) Dataset 25-10-2024



(b) Dataset 26-10-2024

Figure 33

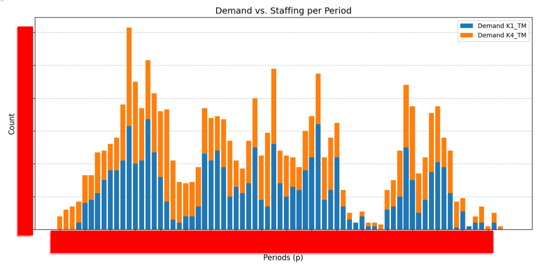


Figure 34: Dataset 27-10-2024