TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Operations Management

Workforce Planning in Manufacturing and Service Industries

Christian Daniel Ruf

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

Doktors der Wirtschaftswissenschaften
(Dr. rer. pol.)

genehmigten Dissertation.

Vorsitzender:		Prof. Dr. Martin Grunow
Prüfer der Dissertation:	1.	Prof. Dr. Rainer Kolisch
	2.	Prof. Dr. Jonathan Bard

Die Dissertation wurde am 27.12.2019 bei der Technischen Universität München eingereicht und durch die Fakultät für Wirtschaftswissenschaften am 15.04.2020 angenommen.

Acknowledgments

I wish to express my gratitude and appreciation to my supervisor Prof. Dr. Rainer Kolisch for his continuous support throughout the completion of my PhD studies. I also want to express my sincere gratitude to Prof. Jonathan F. Bard for the invitation to work with him at UT Austin and for sharing liberally his time and expertise. I am very appreciative for his feedback and ideas, and I enjoyed my time in Austin very much. I also would like to thank Prof. Dr. Grunow for chairing the examination board.

To my former and current colleagues at TUM School of Management: Sebastian Schiffels, Claus Brech, Alexander Döge, Daniel Tonke, Martin Tritschler, Tobias Lieberum, Gregor Godbersen, Maximilian Pohl, Hendrik Weber, Christian Jost, Giacomo Dall'Olio, Pia Ammann, Christine Steinberger, Monika Wagner, Anulark Naber, Pirmin Fontaine, Max Schiffer, Andreas Hottenrott, Mirco Beißner, and Frank Hage: thank you for creating an inspiring work environment, for your support and friendship. Finally, I am grateful to my parents Rita and Jürgen, my brother Michael, and my partner Leah for their limitless support.

Contents

List of Figures

List of Tables

1. Introduction

This dissertation tackles two different managerial problems that relate to the broad field of workforce planning. In the following, I provide a brief overview of workforce planning, and then I specify the topics addressed in this work.

1.1. Workforce planning

Every organization, regardless of its size, requires employees to run the organization and produce its goods and services. Hence, employees are involved in the core functions of every company, and at the same time, are a major cost factor. *Workforce planning* aims at matching the supply and demand for employees; it defines when and how many employees should be hired, promoted, transferred or dismissed and when and where these employees should work, and what they should do (see, e.g. Price et al., 1980; Bruecker et al., 2015). Heterogeneous employees, their preferences, a multitude of workplace regulations, and uncertain personnel requirements make workforce planning one of the most difficult managerial tasks (see Bruecker et al., 2015; Ernst et al., 2004). In theory, an integrated planning approach, in which the above decisions are made simultaneously, generates the best possible planning. However, such a planning problem is intractable. Typically, a hierarchical approach is taken where the planning horizon is decomposed along the time axis giving rise to a series of interrelated problems, often defined as strategic, tactical and operational (e.g., see Koutsopoulos and Wilson, 1987; Gans and Zhou, 2002; Zülch et al., 2004).

Hierarchical planning approaches were first used in production planning, and later applied to a wide range of managerial activities. In his seminal work on planning and control systems, Anthony (1965) was the first to classify managerial activities into three broad categories: (i) long-range strategic planning based on very aggregated data with a high degree of uncertainty, (ii) medium-range tactical planning, and (iii) short-range operational control based on detailed data with a relatively low degree of uncertainty. Different levels of management deal with these three classes of decisions: strategic planning is a responsibility of top management, tactical planning involve middle and top management, and operational control is typically executed by lower management and non-managerial personnel (Silver et al., 2016). Hence, hierarchical decomposition makes the planning problem not only more tractable, but is also aligned with an organization's business practices (Ernst et al., 2004).

Workforce planning at the strategic level is concerned with the long-term development and maintenance of human resources with the appropriate skills to run the organization and produce its goods and services. The mix of employees and their effectiveness are key variables determining an organization's success. Workforce capacity planning corresponds to the long-term capacity planning used in manufacturing environments (Gans and Zhou, 2002). For early reviews of strategic workforce planning, see Price et al. (1980), Purkiss (1981) and Edwards (1983). Bruecker et al. (2015) provide a recent review spanning the whole range of workforce planning, from strategic to operational problems. At the strategic level, the planning horizon is typically two years and longer, and employees are aggregated into homogeneous groups by common characteristics, such as relevant skills, age, and length of service. The selection of these characteristics is domain-specific, balances complexity and accuracy, and also depends on the available data. For example, Fragnière et al. (2010) classify back-office workers in a bank into non-qualified and qualified, the latter of which are costlier and more efficient; Fowler et al. (2008) classify workers in semiconductor manufacturing by their skillsets and general cognitive ability, where the latter influences a worker's productivity, salary and the speed at which new skills are acquired; and Bard and Wan (2008) classify workers in mail processing and distribution centers into full-time and part-time employees and by skill categories. Once the workforce is structured into groups, the next step is to describe how employees transition from one group to another. The means to influence the transitions of employees and thereby the composition of the workforce include hiring, training, promotion, internal transfer, layoff, firing, and retirement planning. Unfortunately, the dynamics are not entirely under the control of the planner. For example, employees may voluntarily leave an organization to relocate, make a career change, or retire. Whether it is manufacturing or service, management's goal is to find the best mix of employees so as to strike a balance between personnel costs and the ability to reliably meet current and future demand for employees. The demand specifies how many employees with which skills are needed at different times and possibly locations over some given planning horizon. Depending on the industry, future demand may be predicted based on tasks (e.g. flight legs requiring cockpit and cabin crew, Cordeau et al., 2000) or workload forecasts (e.g call centers, Gans and Zhou, 2002) or simply from a minimum number of required workers to operate a facility (e.g. chemical plants, see Chapter 2). Poor demand predictions or capacity planning can lead to an oversupply of workers with too much idle time, or periods with an undersupply of workers with an attendant loss of business (Bard and Wan, 2008). In spite of good planning, when demand or capacity fluctuate it is often impossible to perfectly match the available skill pool to the demand. Besides overtime and employing temporary workers, a popular option to mitigate this is to increase worker flexibility by means of cross-training (e.g. Campbell, 2011; Fowler et al., 2008) or substitution (e.g. Bordoloi and Matsuo, 2001; Bard and Purnomo, 2005). *Cross-training* refers to extending workers' skillsets by teaching them additional skills. When skills are hierarchically structured, higher skilled workers may fill in for lower skilled workers when shortages exist, which is known as *substitution* or *downgrading*. Both cross-training and substitution can provide substantial cost savings. Pinker and Shumsky (2000) investigate whether cross-training while enhancing flexibility can create a workforce that performs many tasks with consistent mediocrity, and come to the conclusion that the optimal staff mix combines flexible and specialized workers. A related question is whether to hire predominantly specialists and broaden their horizons or predominantly generalists and give them specific training (Price et al., 1980).

These strategic decisions determine the composition of the permanent workforce and thus constrain the lower planning levels. At the tactical level, assuming a permanent workforce, timetables must be constructed such that operational requirements can be met. This process is called *rostering* or *personnel scheduling*. For reviews of personnel scheduling, see Ernst et al. (2004), Alfares (2004) and Van den Bergh et al. (2013). There is also a large body of research with respect to nurse rostering (see Burke et al., 2004, for a review). In personnel scheduling, the planning horizon is typically from half a year to two years. When the length of a workweek does not match the operating week, personnel scheduling must include *days-off scheduling*, i.e. determining an employee's work- and rest-days. When the length of an employee's workday does not match the length of daily operations, personnel scheduling must include *shift scheduling*, i.e. determining the shift for each workday of an employee. An integrated approach that addresses both aspects is *tour scheduling* (Baker, 1976; Brusco and Jacobs, 1998), which involves the creation of either cyclic or non-cyclic schedules. A *cyclic schedule* consists of a set of work patterns which are rotated among groups of workers, such that each group completes each pattern exactly once per iteration. Cyclic schedules are fair, offer a high degree of regularity and require relatively low scheduling effort. However, they lack flexibility when reacting to demand fluctuations or absences due to, say, illness or vacations (Millar and Kiragu, 1998), and they do not take into account seniority or individual preferences. Kiermaier et al. (2016) introduce the concept of a flexible cyclic roster that offers many of the benefits of non-cyclic schedules while maintaining a high degree of regularity and fairness. A recent trend in rostering, especially in industries where employee satisfaction is key, is to take individual preferences into account. The motivation for *preference scheduling* is that more individual control over workweeks will lead to higher morale, a more attractive work environment, increased flexibility to deal with personal matters, and higher retention rates (Bard and Purnomo, 2005). In general, rostering tends to become complex due to the need to account for a variety of workplace regulations, especially when incorporating measures of fairness and employee satisfaction (see Ernst et al., 2004; Balakrishnan and Wong, 1990).

At the operational level, assigning tasks to individual shifts or individual employees may be necessary. A task usually has a start time or a time window and a duration and possibly a spacial dimension, where the start and end locations may or may not be identical. Furthermore, a task may require one or more employees with certain skills and additional equipment. For example, Eiselt and Marianov (2008) assign employees to tasks, minimizing the discrepancy between an employee's abilities and the task's requirements, minimizing overtime and unfair distribution of the workload, in order to increase job satisfaction. For that purpose, they specify the skill requirements and employees' skills in a multi-dimensional skill space (Eiselt and Marianov, 2008). In many cases, tasks can be executed in different ways, giving rise to multiple modes, which allow for several kinds of trade-offs, such as time/resource, time/cost and resource/resource trade-offs (Reyck and Herroelen, 1999). One area with many modes is airport baggage handling, where the baggage of each departing flight needs to be loaded into containers. While the end time is determined by the flight departure, the start of the loading process and how many workers should be loading bags at different times during the loading process, are flexible (multiple modes). By planning the loading process, the speed at which bags are loaded and the utilization of the airport facilities can be controlled (resource/resource trade-offs).

A major challenge in workforce planning is uncertainty, which is ubiquitous at all planning levels (see, e.g. Price et al., 1980). Uncertainty can be found on both demand- and supply-side. On the demand-side, uncertainty exists with regard to the tasks (quantity and duration) or the number of incidents requiring employee involvement (e.g. inbound calls in a call-center, emergencies in a hospital, checked-in bags). On the supply-side, the availability of employees may fluctuate. The central source of uncertainty at the strategic level is the rate of voluntary turnover (Edwards, 1983). At the tactical and operational level, absence, for example due to illness or parental leave, can cause unexpected capacity shortfalls (e.g. Fragnière et al., 2010). Another example where capacity is reduced on short notice is in crew rostering in the airline industry. Assume that there is congestion at the destination airport, which requires additional flying time for the cockpit crew. By the time the aircraft lands, the crew may not be legally allowed to fly their subsequent leg because they have exceeded their allowed flying time, rendering a disruption to the subsequent legs (Petersen et al., 2012). Approaches that address uncertainty, such as stochastic programming, Markov decision problems or robust optimization, can be expected to outperform deterministic approaches. Nevertheless, Purkiss (1981) observed in his practically oriented review of manpower planning models that linear programming, despite using deterministic data, is more popular than Markov decision problems, which can handle uncertainty. In fact, this trend has continued: the number of workforce planning related papers that incorporate uncertainty in their model remains rather limited compared to the papers that investigate deterministic problems (Bruecker et al., 2015). The reason for the lack of research on this subject is the additional computational complexity, introduced by the explicit consideration of uncertainty, which prohibits the use of such techniques in many cases (see Price et al., 1980).

1.2. Structure of the dissertation

This dissertation tackles a strategic workforce planning problem in Chapter 2 and an airport operations problem, in which workers are allocated to flights, in Chapter 3. Both problems involve uncertainty, which is explicitly considered in the optimization models. Since the models are difficult to solve, customized solution procedures have been developed for each context.

Chapter 2 uses results from the working paper Ruf et al. (2019a), which was submitted for publication to the Production and Operations Management Journal on the 26th of November 2018, and which is currently being revised. I am the main contributor to all parts of the working paper from posing the research question, development of the model, conceptual development and implementation of the solution procedure, design of the computational study and its execution, and analysis and interpretation of the results.

This chapter addresses a strategic workforce capacity planning problem for a hierarchically skilled workforce in a production environment, focused on hiring and training policies in the face of random resignations. Recruits are hired with little or no experience and are trained over multiple periods to perform jobs that require ever greater skills. The training can take place either off-the-job, on-the-job or a combination thereof. The problem is complicated by random resignations that can lead to labor shortfalls that jeopardize continuous operations. The objective is to balance workforce costs (salaries, hiring and training) with penalty costs associated with skill shortages. The problem is modeled as a Markov decision problem for which several parameterized decision rules are proposed to find solutions. Good parameter values are determined with a very large-scale neighborhood search that is designed to deal with noisy cost function measurements. My computational experiments show that good parameter values can be found in less than four hours for instances derived from real-world data. When training requires extensive supervision, my results indicate that the number of workers concurrently in training should be limited. They also show that a shorter, intense training period during which employees do not perform regular tasks is preferable, in general, to a longer training period where employees spend time both on and off the job. Finally, the results demonstrate the value of worker flexibility when downgrading is allowed.

Chapter 3 of this dissertation uses results from the working paper Ruf et al. (2019b), which was submitted for publication to the journal Transportation Science on the 7th of November 2019. I am the main contributor to all parts of the working paper from posing the research question, development of the model, conceptual development and implementation of the solution procedure, design of the computational study and its execution, and analysis and interpretation of the results.

This chapter deals with the intra-day planning of an airport's outbound baggage handling processes. Outbound baggage is transferred from the terminal or transfer flights to departing airplanes. To that end, each flight's baggage has to be loaded into containers, which will then be forwarded to the airplane. Planning the loading process consists of setting the start times for the loading process and depletion of the baggage storage, as well as assigning handling facilities and workers. Flight delays and uncertain arrival times of passengers at the check-in counters require robust plans that are adjusted dynamically every few minutes, and hence necessitate an efficient planning procedure. The problem is modeled as an integer program that utilizes observed flight data to generate robust plans in a rolling planning fashion, allowing problem parameters to be updated in each re-optimization. A column-generation-based heuristic is employed to solve the problem in limited computational time. My computational experiments demonstrate the performance of the proposed procedure based on real-world data from a major European airport. The results show that (i) the procedure outperforms a heuristic that mimics manual decision-making, and (ii) being able to dynamically (re-)allocate baggage handlers leads to improved solutions with considerably fewer mishandled bags. Finally, Chapter 4 draws conclusions from the previous two chapters.

2. Workforce Capacity Planning with Downgrading, Training, and Resignations

2.1. Introduction

Strategic workforce planning is concerned with the long-term development and maintenance of employees with the appropriate skills to run the organization and produce its goods and services (see Price et al., 1980). This involves five tasks: (i) identifying the relevant skill types and structures, (ii) determining the size and skills of the current workforce, (iii) predicting the future supply of qualified job-seekers, (iv) predicting the future demand for each skill, and (v) determining a capacity plan to develop and maintain an appropriate workforce (see Edwards, 1983; Song and Huang, 2008). This chapter focuses on (v), where the workforce is hierarchically skilled and fluctuates over time.

Bruecker et al. (2015) identify two skill classes common in the literature: categorical and hierarchical skills. In case of categorical skills, each worker has a set of skills and must possess the skills required for a task in order to be able to perform it. For hierarchical skills, workers with a higher skill level are more educated or have more experience and can therefore perform more tasks, or they can perform certain tasks better or faster (Bruecker et al., 2015). Some authors combine hierarchical and categorical skills to better capture the skill structure of the organization under consideration. For example, Wirojanagud et al. (2007) and Fowler et al. (2008) use categorical skills and additionally place workers on a hierarchical scale measuring a worker's cognitive ability, which influences their productivity, costs, and learning speed. When skills are hierarchically structured, higher skilled workers may fill in for lower skilled workers when shortages exist. This is referred to as *substitution* or *downgrading* (Emmons and Burns, 1991; Bard, 2004; Bard and Purnomo, 2005) and adds flexibility, but excessive use may be perceived as demeaning and lead to boredom, dissatisfaction, and eventually to lower retention rates (Eiselt and Marianov, 2008).

There are different ways of establishing and managing a permanent workforce: recruiting, internal transfers, training, management of turnover, retirement planning and layoffs being the key instruments for controlling the capacity. Increasing the total size of the workforce requires recruiting. When skills are hierarchical, although it may be possible to hire experienced employees, it is often the case that new employees start at the lowest skill level (e.g., Gans and Zhou, 2002; Wang, 2005; Bordoloi and Matsuo, 2001; Anderson, 2001; Sandborn and Prabhakar, 2015). To increase the overall skill level of a workforce, either learning or training is required. While *learning* happens automatically through repeating tasks, acquiring certain skills or licenses necessitates formal training. Training can be conducted off-the-job or on-the-job or in some combination of the two. Although employees in training do not contribute fully (if at all) to capacity and may even require that instructors be temporarily removed from the workforce, reduced availability during training is rarely considered in the literature (Bruecker et al., 2015). However, training can consume a considerable amount of time, in which case capacity reductions are significant.

Unfortunately, the dynamics are not entirely under the control of the planner. Employees may voluntarily leave an organization, for example, for family relocation, a change in career, or retirement. The rate of voluntary turnover, being the key uncontrolled factor (Edwards, 1983), makes workforce capacity planning one of the most challenging managerial tasks. New recruits in particular exhibit high turnover rates since many find that their aptitudes and preferences are not a good fit for the job (Bordoloi and Matsuo, 2001). Experienced employees on the other hand are less likely to resign, but they are much harder to replace when they do. The situation becomes problematic when unexpected resignations lead to capacity shortfalls. When there is a lack of skilled employees, hiring and training must be given top priority despite the accompanying drop in manpower. Resignations combined with long training durations and reduced availability during training require that somewhat larger workforce levels be maintained in many production environments to prevent daily disruptions.

When building and maintaining a workforce, an organization must strike a balance between personnel costs (hiring and training costs, salaries) and the negative consequences of potential capacity shortfalls. Personnel costs increase with the size of the workforce, and are a result of strategic workforce planning decisions. Capacity shortfalls occur when the workforce is too small. They are most evident when making medium- to short-term personnel scheduling decisions that involve the construction of weekly timetables, and short-term operational planning, such as the daily assignment of tasks.

The planning problem tackled in this chapter derives from the steel industry, where highly specialized workers control and monitor processes in plants that operate 24/7. Common examples include the production of pig iron in blast furnaces, and the further processing in steel and rolling mills. A furnace may operate continuously over a period that can exceed 20 years (e.g., see Thyssenkrupp AG, 2014). To keep production on track, workers with ever increasing skills are required at the melting range, at the pig iron platform, and at the control center. As the smelting process is subject to variation, constant monitoring and adjusting are required to prevent costly disturbances. The vocational training to become a steelworker alone takes 3.5 years. Trainees are paid a monthly salary and receive instructions in the classroom and in the plant. Having passed their exam, trainees become assistants. Most assistants are directly employed by the company that trained them. Subsequently, an assistant can undergo further training to become a foreman in metallurgy, which takes about three additional years. Before an employee can occupy a position in the control center further supervision by experienced co-workers is necessary. As steelworkers are in great demand, the pool of qualified workers is severely limited, so companies have to rely on building a workforce with their own apprentices.

Another area where this problem exists is the chemical industry, where basic chemicals such as methanol or ammonia are processed by specialized workers running highly automated equipment and plants. Such plants operate continuously and require a minimum number of licensed operators to keep them going. If for some reason there is a shortage of operators at any position, the plant has to be shut down, which leads to lost production and a host of costs related to the shutdown and subsequent restart. The licenses to operate a plant at a certain skill level are obtained in successive stages. The apprenticeship to become a *chemical plant operator* alone takes three and a half years. All further licensing is plant-type specific, so except in rare instances, it is not possible to hire skilled employees. Hence, deliberate workforce planning is essential in both industries mentioned.

The purpose of this chapter is to investigate the strategic workforce capacity planning problem with a hierarchically skilled workforce, downgrading, extended training periods during which employee productivity is reduced, and random resignations. The goal is to develop hiring and training policies that balance workforce costs with the negative consequences of under-staffing. The problem is modeled as a discrete-time, infinite horizon Markov decision process with discrete states (i.e., numbers of workers) and stationary data, including training durations and costs, salaries, demands, and resignation probabilities. This approach makes it easy to include random resignations. The assumption of stationary demands is reasonable as the facilities targeted are long-term investments and run continuously for many years. Stationary data further allows for employing time-invariant and simple decision-rules suitable for strategic decision-making. The decision rules are of a "base-stock" type common in inventory literature. However, since training reduces capacities in the short run, a simple base-stock rule does not perform well. Consequently, two enhancements are proposed: a "base/minimum-stock" decision rule and a "base/minimum-stock/training-capacity" decision rule. For the latter, each decision is made as a function of the current state, base-stock, minimum-stock as well as training-capacity values, which determine the desired number of workers, the minimum number of workers, and the maximum number of workers that can be in training simultaneously for each skill level, respectively. The challenge is to find good values for the decision rule parameters in the presence of noisy measurements. To address this issue, I present a very large-scale neighborhood search that quickly finds the promising region of the solution space and then increases the sample size in that region. Computational experiments demonstrate the effectiveness of the proposed approach in solving instances derived from real world problems. They further indicate that when on-the-job training does not require excessive supervision, decisions of the base/minimum-stock type perform well.

The contributions of this chapter are threefold. First, a new model for the strategic workforce planning problem with downgrading, extensive training requirements and random resignations is provided; second, intuitive and effective parameterized decision rules to find solutions are provided; and third, an effective search procedure to find good parameter values is developed. Although there is an abundance of literature on workforce planning, determining the precise timing of hiring and training while accounting for reduced capacity when trainees and instructors are removed from the shop floor is a surprisingly difficult problem due to its combinatorial nature (see Bruecker et al., 2015). These issues are addressed here and managerial guidance on preferable training schedules and on the value of downgrading are provided. While both mentioned applications can be categorized as continuous manufacturing industries, where the main processing is typically done at facilities that have very high startup and shutdown costs and work around the clock (see Kreipl and Pinedo, 2004), the model is applicable to different industries as long as skills are strictly hierarchical. For example, Gans and Zhou (2002) apply their approach to call centers. Their model differs from ours in that workers progress further up in the skill hierarchy automatically through learning by doing, instead of formal training that is initiated by the planner.

The remainder of this chapter is structured as follows. Section 2.2 provides an overview of strategic workforce planning and clarifies its relationship with inventory problems. Section 2.3 describes the tackled problem and assumptions made in the model in detail, and Section 2.4 introduces the Markov decision process model. Section 2.5 develops the decision rules and the neighborhood search heuristic. Section 2.6 contains the computational experiments, followed in Section 2.7 with insights and conclusions drawn from the study.

2.2. Literature Review

The first part of this section introduces concepts from the field of workforce capacity planning, the second part reviews research related to the problem of this work. The third part highlights the relationship between this work and multi-echelon inventory problems.

Workforce capacity planning. As mentioned, categorical and hierarchical skill structures, and combinations thereof are common in literature. In the context of this work, skills are hierarchical. Hierarchical skills can be defined on a continuous scale or at discrete levels depending on what the skills represent and how they change over time. For example, when experience increases with age a continuous scale is appropriate. Skills acquired in training, as in this work, can be suitably expressed in discrete increments. There are different ways of establishing and managing a permanent workforce. The main instruments for the planner to adapt the skill pool are recruiting, career planning, promotions, internal transfers, training, management of turnover, and retirement planning (see

Price et al., 1980). Although it may be possible to hire experienced employees, it is often the case that new employees start at the lowest skill level (e.g., Gans and Zhou, 2002; Wang, 2005; Bordoloi and Matsuo, 2001). To increase the overall skill level of a workforce, learning or training is required. *Learning* happens automatically by repeating tasks. On the other hand, employees may loose knowledge if a task is not performed for a long time, which is referred to as *forgetting* (Bailey, 1989; Valeva et al., 2017). Learning may occur at different rates, which is known as the *learning curve.* Learning is sometimes considered a random variable in workforce planning research when the planner cannot exercise direct control and because of the large variance in the rate at which apprentices learn (see, e.g. Anderson, 2001; Gans and Zhou, 2002). Formal training is a more direct way at the planner's disposal to substantially raise an employee's skill level. Training often incurs a cost and can be conducted off-the-job or on-the-job. When employees undergo off-the-job training they do not contribute to production; during on-the-job training the trainee may contribute to production, but at the same time a supervisor out of the workforce may be required, which would fully or partially remove him from performing his normal functions. The overall size of the workforce decreases, when employees leave an organization, which may be initiated by the employer, as in the case of layoffs, or by the employee, as in the case of resignations. Whether companies should lay off experienced employees during business downturns is a question investigated by Anderson (2001). *Critical skill loss* describes the loss of skills that are either non-replenishable or take very long periods of time to reconstitute (Sandborn and Prabhakar, 2015). In the context of this work, employees voluntarily resign and training takes considerable time, e.g. years.

Related work. Martel and Price (1981) deal with a multistage workforce capacity planning problem extending previous approaches to cover the case where manpower demands are uncertain. Their model is very general in that workers can be in arbitrary states. However, they do not consider random resignations. Anderson (2001) investigates a similar problem with two skill levels: unproductive apprentices and fully productive experienced employees, and non-stationary stochastic demand for a single product or service. Changes to the workforce occur when trainees become experienced employees and when experienced employees retire, which are assumed to occur at constant rates. The employer can adjust the workforce by hiring apprentices and laying off experienced employees in order to balance the costs and time lags associated with training against the need to meet ever changing demand. The model proves useful for understanding the strategic dynamics of workforce capacity planning in a volatile environment. In contrast, in our model workers advance through multiple skill levels as a result of training decisions made in a stationary environment but in light of stochastic resignations.

Bordoloi and Matsuo (2001) consider a workforce capacity planning problem related to staffing an assembly line of a factory producing semiconductor equipment. Two different skill levels are required on the line, while employees at a third level can train and substitute for those at the lower levels. The demand for the final product is an independent and identically distributed random variable. Fixed training and retention rates per period are assumed, and the employer is allowed to adjust the composition of the workforce by hiring recruits at discrete points in time.

Gans and Zhou (2002) investigate a multistage workforce capacity planning problem with a discrete skill hierarchy and random non-stationary demand. In the model, planning is done at discrete points in time and inventory levels are treated as continuous. Additions to the workforce are permitted solely by hiring unskilled recruits. At the end of each period the workers at each skill level randomly either quit, progress to the next skill level, or stay at their current level. Different patterns are possible at different levels and in different periods. The authors show that a “hire-up-to” type policy is optimal. They also develop an LP-heuristic based on average learning and turnover rates, and find that the heuristic does not recognize the need for a “buffer” of excess staff. Therefore, when additional capacity is expensive or unavailable, explicit modeling of stochastic learning and turnover effects may improve performance significantly. Our model is similar to that of Gans and Zhou (2002) in that both capture the interdependence of hierarchical workforce planning with the help of a cost function that can be adapted to the operational environment. Instead of focusing on optimal hiring policies and learning effects, though, we consider long-term training. Ahn et al. (2005) extend the work of Gans and Zhou (2002) to discrete inventory levels.

Song and Huang (2008) tackle a multistage workforce capacity planning problem with categorical skills (representing business units) and random turnover. As is typically the case, the skill pool is adjusted by hiring, firing and transferring workers between business units. Each of these actions incurs a cost and may be bound by company policies. The authors model the problem as a time-space network and apply a successive convex

<table>
<tr><th></th><th>Martel</th><th>Anderson</th><th>Bordoloi</th><th>Gans</th><th>Song</th><th>Ruf</th></tr>
<tr><td>Hierarchical skills</td><td rowspan="6">possible (arbitrary states)</td><td>2 levels</td><td>3 levels</td><td>many levels</td><td>no</td><td>many levels</td></tr>
<tr><td>Demand per skill</td><td>no</td><td>no</td><td rowspan="2">possible</td><td>yes</td><td>yes</td></tr>
<tr><td>Downgrading</td><td>no</td><td>yes</td><td>no</td><td>yes</td></tr>
<tr><td>Hiring decision</td><td>yes</td><td>yes</td><td>yes</td><td>yes</td><td>yes</td></tr>
<tr><td>Training decision</td><td>no</td><td>no</td><td>no</td><td>yes</td><td>yes</td></tr>
<tr><td>Long-term training</td><td>no</td><td>no</td><td>no</td><td>no</td><td>yes</td></tr>
<tr><td>Random resignations</td><td>no</td><td>no</td><td>no</td><td>yes</td><td>yes</td><td>yes</td></tr>
</table>

Table 2.1.: Literature summary

approximation method to obtain near-optimal solutions. In contrast, we use a Markov decision model with hierarchical skills in which training durations may span more than a single period. Table 2.1 summarizes the features of the related works and compares them to the problem presented here.

Relationship to inventory models. A similarity exists between our problem and multi-echelon inventory problems with a serial product structure, a single product, stationary problem data, and periodic reviews (see Clark, 1972). The inventory at each level relates to the available workers and holding costs equate to salaries. Moreover, order lead times correspond to training durations, ordering costs correspond to training costs and random resignations correspond to state-dependent demands or product spoilage. What distinguishes the problem considered here is that there is internal demand for manpower at each skill level that can be satisfied by employees qualified to work at that level or (due to the possibility of downgrading) at a higher level. In case demand cannot be met shortage costs are incurred. In addition, employees in training from one skill level to the next higher level affect capacity at the next higher level and at the current level. If an employee needs a supervisor during on-the-job training, “shipping” an employee may temporarily reduce the available capacity at the next higher level, and if training does not fully utilize a worker, he contributes to capacity at the current level at a rate equal to the fraction of remaining time. These factors add several layers of complexity to the problem at hand. The decision rules presented in Section 2.5 are similar to a base-stock policy, also commonly referred to as an $(S-1, S)$, order-up-to, or one-for-one replacement policy (see, e.g., Scarf, 1960; Clark and Scarf, 1960). Using a simple base-stock policy, new workers are requested as soon as the inventory position falls below a given level so as to restock up to that level. Nevertheless, because sending workers to training reduces capacity at the starting level and may additionally decrease capacity at the level that requests new employees, a more careful order and supply policy is needed.

2.3. Problem Description

The objective of the planning is to maintain an appropriately skilled workforce in the face of random resignations and given demand. The options for doing so include (i) hiring unskilled recruits, (ii) training employees from one skill level to the next over an extended period of time, and (iii) downgrading workers for any length of time. Planning decisions are made at discrete points in time, here every six months, and training can take up to three and a half years. Between decision points, the workforce composition is assumed to be constant. In each period and for each skill level, costs are incurred when the workforce is insufficient to cover the demand. Costs include overtime, and in the extreme, plant shutdowns.

Hiring. Steelworkers and others in the process industries are in great demand, and virtually all recruits continue to work at the company where they are trained. Because it is rarely possible to hire skilled workers in these industries, companies have to rely on building a workforce with their own apprentices. Therefore, it is assumed that all entrants are unskilled and are immediately enrolled in an apprenticeship program that spans a minimum of one period.

Training. Training raises a worker's current skill level to the next higher level and requires one or more planning periods of non-interruptible participation. Both on-the-job and off-the-job options are common. While a worker is undergoing training, his availability at his current skill level may be reduced. During off-the-job training no work is performed. During on-the-job training a worker is not available at his current skill level, but may contribute to production at the skill level he is being trained for, which increases capacity at that level. On the other hand, an instructor may be required for supervision, which reduces capacity at that level. Hence, the net contribution at the target level may be positive, or negative when the capacity gained by having the apprentice do tasks is less than the capacity lost from diverting part of the supervisor's time from production to training (Anderson, 2001). While training schedules may differ from level to level, it is assumed that they are identical for all employees at the same skill level. In practice, off-the-job training is more prevalent at the beginning of the period while on-the-job training is more prevalent towards the end. For simplicity we assume that the proportion of time spent off- and on-the-job training remains constant throughout the training period, although the corresponding ratio can be a function of the level. Nevertheless, the model can be easily adapted to allow for dynamic adjustments of these proportions.

Resignations. Typically, resignations occur randomly and are individually motivated (Gans and Zhou, 2002). They can adequately be modeled with a binomial distribution in each period and at each skill level, assuming a known probability that depends on the skill level. This allows us to account for higher resignation rates among new recruits. It is further assumed that employees who hand in their resignation during a planning period actually leave the workforce at the end of that period, a reflection of the fact that most worker contracts have a clause that requires a certain amount of notice. Also, we assume that workers do not resign while in training. This can be justified by noting that professional development usually increases job satisfaction and reduces turnover (Eiselt and Marianov, 2008). From a modeling and computational point of view, this assumption helps reduce complexity and simplifies the presentation without sacrificing much in accuracy.

Retirements. Unlike resignations, retirements are in general known well in advance, and it is a comparably easy task to replace retiring workers by hiring and training new employees in a timely manner. Not considering retirements means not considering the capacity a newly hired worker contributes and the costs he causes while he is in training to fill the vacancy created by a retiring worker. Since, including this aspect does not influence the dynamics much, but would not allow to tackle the problem as a stationary Markov decision process, it is not included in the model.

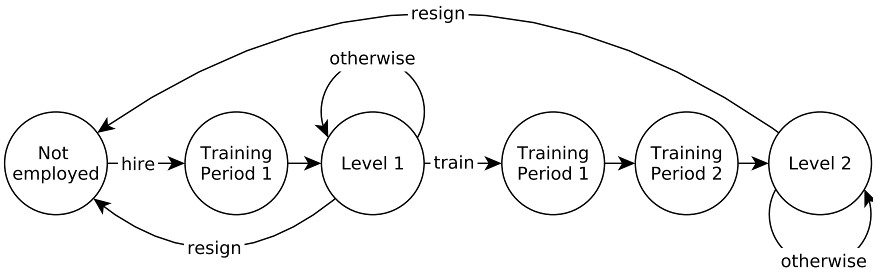


Figure 2.1.: Employee state diagram

Figure 2.1 depicts the state diagram for one employee with two skill levels. The first round of training starts immediately and takes two periods; three more periods are required to achieve proficiency at level 2. An employee can resign at any level but not while in training.

Costs. To keep a plant operating smoothly, a minimum number of operators at each skill level is required. Workers at higher skill levels can be downgraded to do jobs that are usually done by lower skilled workers, which provides some flexibility. A lack of operators can lead to reduced output and/or poor quality as well as costly disruptions. At a furnace, for example, operators must continuously control the smelting process. Often, in case of worker shortages, corrective measures must be taken immediately to avoid disastrous situations such as "hanging burden" or "scaffolding" (Sarna, 2013).

In the model, labor requirements are given in terms of the minimum numbers of workers per skill level necessary to operate a plant continuously over the course of one planning period. These requirements could be derived by solving a single personnel scheduling problem with demand per level as input. To focus on strategic planning, however, the minimum worker demands are taken as given (refer to Bard (2004); Billionnet (1999); Al-Yakoob and Sherali (2007) to see how the actual values might be derived). Staff shortages are penalized in the model. Available capacity per skill level can be distributed among the same and all lower skill levels, and a penalty is incurred per unit of missing capacity. In practice, minor personnel shortages can be compensated with overtime. Regulations regarding overtime pay make the labor cost a piecewise linear convex function of the required capacity (e.g., see Ye et al., 2018). When overtime does not suffice to compensate for the shortage, operations are endangered. Computational experience has shown that my methodology can handle different cost functions. For ease of presentation, though, linear penalty costs are assumed.

Additionally, each worker receives a salary based on his skill level. Salaries may deviate for workers in training, and training may incur additional costs. The objective is to minimize the expected infinite sum of discounted costs per period, consisting of personnel costs and penalties for staff shortages given the current workforce composition.

2.4. Model

In this section, the workforce capacity planning problem is modeled as a discrete-time, infinite horizon Markov Decision Process (MDP) with planning horizon $\mathcal{T} = \{0, \ldots, \infty\}$. The notation used to present the model is summarized in Table A.1 of Appendix A. At any time, each employee is at one of L skill levels in the set $\mathcal{L} = \{1, 2, \ldots, L\}$. Changes to the workforce composition can only take place at fixed points in time. Dummy skill level 0 represents an external placeholder from which workers are hired. As mentioned, new recruits have to undergo initial training. For simplicity, we may refer to hiring as training from level 0. Let $\mathcal{L}_0 = \{0, 1, \ldots, L-1\}$ denote the set of skill levels from which workers can start training, and let Δ_l denote the training duration in planning periods to proceed from Level $l \in \mathcal{L}_0$ to $l+1$.

2.4.1. States

For each possible training start time, the pre-decision state comprises (i) the number of workers at each skill level, and (ii) the number of workers who are in training at each level. Variables $z^{\mathrm{a}}_{tl} \in \mathbb{N}_0$ track the number of available workers at level $l \in \mathcal{L}$ at time $t \in \mathcal{T}$, excluding workers currently in training. Variables $z^{\mathrm{t}}_{tlt'} \in \mathbb{N}_0$ track the number of workers at level $l \in \mathcal{L}_0$ in training at time t to reach level $l+1$ who started their training at time $t' \in \mathcal{T}_{tl}$, where $\mathcal{T}_{tl} = \{t - \Delta_l + 1, \ldots, t-1\}$. Variable vector $z_t \in \mathbb{N}_0^K$ with $K = \sum_{l \in \mathcal{L}_0} \Delta_l$ denotes the pre-decision state at time $t \in \mathcal{T}$. Furthermore, $\mathcal{Z} \subseteq \mathbb{N}_0^K$ denotes the state space.

To illustrate the model, the following problem instance is used throughout this section.

- Two skill levels: $L = 2$
- Initial training duration: $\Delta_0 = 2$
- Training duration from level 1 to level 2: $\Delta_1 = 3$

The state at $t \in \mathcal{T}$ is given by $z_t = \left(z^{\mathrm{a}}_{t,1}, z^{\mathrm{a}}_{t,2}, z^{\mathrm{t}}_{t,0,t-1}, z^{\mathrm{t}}_{t,1,t-2}, z^{\mathrm{t}}_{t,1,t-1}\right)$. Variables $z^{\mathrm{a}}_{0,1}$ and $z^{\mathrm{a}}_{0,2}$ are the number of available level 1 and 2 workers at time $t = 0$, respectively. Variable $z^{\mathrm{t}}_{0,0,-1}$ is the number of workers who have been hired one period ago at time -1. They will finish their training at the end of the current period at time 1. Variables $z^{\mathrm{t}}_{0,1,-2}$ and $z^{\mathrm{t}}_{0,1,-1}$ are the numbers of workers who started training at level 1 to reach level 2 at times -2 and -1, respectively; $z^{\mathrm{t}}_{0,1,-2}$ workers will become available at level 2 at the end of the current period.

2.4.2. System dynamics

The evolution of the system from one decision to the next is a two-step process (see Figure 2.2). The first step captures the immediate effects of the decisions on the workforce leading from the pre-decision state to the post-decision state. The one-period costs are calculated based on the post-decision state as explained below. The second step is the transition from the post-decision state to the pre-decision state of the next period. This separation requires defining post-decision state variables, but simplifies the exposition.

Decisions. Let decision variables $x_{tl} \in \mathbb{N}_0$ be the number of workers currently at level $l \in \mathcal{L}_0$ sent to training at time $t \in \mathcal{T}$. Let $x_t = (x_{tl})_{l \in \mathcal{L}_0} \in \mathbb{N}_0^L$ denote the decision vector at time t. The feasible region is defined by the following constraints.

$$x_{tl} \leq z^{\mathrm{a}}_{tl} \qquad \forall t \in \mathcal{T}, l = 1, \ldots, L-1 \tag{2.1}$$

$$x_{tl} \in \mathbb{N}_0 \qquad \forall t \in \mathcal{T}, l \in \mathcal{L}_0 \tag{2.2}$$

Constraints (2.1) ensure that no more than the available level l workers are trained and constraints (2.2) define the variable domains.

The post-decision state variables track the workforce composition directly after hiring and training. They are signified by the superscript "x". Variables $z^{\mathrm{ax}}_{tl} \in \mathbb{N}_0$ represent the

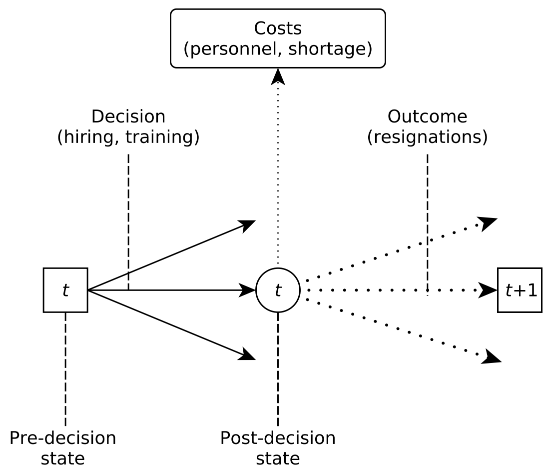


Figure 2.2.: Sequence of decisions, cost calculations and outcomes

number of available workers at level $l \in \mathcal{L}$, and variables $z^{\mathrm{tx}}_{tlt'}$ represent the number of level $l \in \mathcal{L}_0$ workers who are in training to reach level $l+1$ and started their training at $t' \in \mathcal{T}_{tl} \cup \{t\}$. Similarly as before, variables $z^{\mathrm{x}}_t \in \mathbb{N}_0^K$ with $K = L + \sum_{l \in \mathcal{L}_0} \Delta_l$ denote the post-decision state of the workforce at time $t \in \mathcal{T}$. The transition function from pre-decision state z_t to post-decision state z^{x}_t, given the decision vector x_t, is denoted by $z^{\mathrm{x}}_t = S^{\mathrm{Mx}}(z_t, x_t)$ and defined by the following constraints.

$$z^{\mathrm{tx}}_{tlt} = x_{tl} \qquad \forall t \in \mathcal{T}, l \in \mathcal{L}_0 \tag{2.3}$$
$$z^{\mathrm{ax}}_{tl} = z^{\mathrm{a}}_{tl} - x_{tl} \qquad \forall t \in \mathcal{T}, 1 \leq l \leq L-1 \tag{2.4}$$
$$z^{\mathrm{ax}}_{tL} = z^{\mathrm{a}}_{tL} \qquad \forall t \in \mathcal{T} \tag{2.5}$$
$$z^{\mathrm{tx}}_{tlt'} = z^{\mathrm{t}}_{tlt'} \qquad \forall t \in \mathcal{T}, l \in \mathcal{L}_0, t' \in \mathcal{T}_{tl} \tag{2.6}$$

Constraints (2.3) track the newly hired or currently available workers who are sent to training. Constraints (2.4) remove the workers sent to training from the available pool. All other state variables remain unchanged as indicated by constraints (2.5) and (2.6).

Extending the previous example, assume the state at time 0 is such that

- one worker A is available at level 1 ($z^{\mathrm{a}}_{0,1} = 1$)
- one worker B is available at level 2 ($z^{\mathrm{a}}_{0,2} = 1$)
- one worker C is in training from level 1 to 2, and has started training at time -2

($z^{\mathrm{t}}_{0,1,-2} = 1$)

Assume that the decision at time 0 is such that one worker D is hired ($x_{0,0} = 1$) and that worker A is sent to training ($x_{0,1} = 1$). Then, the post-decision state $z^{\mathrm{x}}_0 = S^{\mathrm{Mx}}(z_0, x_0)$ becomes

- worker D is in initial training ($z^{\mathrm{tx}}_{0,0,0} = x_{0,0} = 1$)
- worker A is in training from Level 1 to 2 ($z^{\mathrm{tx}}_{0,1,0} = x_{0,1} = 1$) and not available anymore ($z^{\mathrm{ax}}_{0,1} = z^{\mathrm{a}}_{0,1} - x_{0,1} = 1 - 1 = 0$)
- worker B is available at Level 2 as before: $z^{\mathrm{ax}}_{0,2} = z^{\mathrm{a}}_{0,2} = 1$
- worker C continues training ($z^{\mathrm{tx}}_{0,1,-2} = z^{\mathrm{t}}_{0,1,-2} = 1$)

Transition to the next period. When transitioning from post-decision state z^{x}_t to the next period's pre-decision state z_{t+1} two things happen. First, during period $(t, t+1)$ employees at any skill level may hand in their resignation. It is assumed that those workers continue working until the end of the period, and actually leave the company at time $t + 1$. The probability that a level l worker resigns is denoted by p^{r}_l. Binomially distributed random variable $X^{\mathrm{r}}_{tl} \sim B\left(z^{\mathrm{ax}}_{t-1,l}, p^{\mathrm{r}}_l\right)$ gives the number of workers that leave at time $t \in \mathcal{T}$ under the assumption that departures are independent. For convenience, the vector of random variables $X^{\mathrm{r}}_t = (X^{\mathrm{r}}_{tl})_{l\in\mathcal{L}}$ is defined. Second, workers in training either finish their training at the end of the period if the training period is over, or remain in training. The transition function from post-decision state z^{x}_t to the next pre-decision state z_{t+1}, given the random information X^{r}_{t+1}, is denoted by $z_{t+1} = S^{\mathrm{MX}}\left(z^{\mathrm{x}}_t, X^{\mathrm{r}}_{t+1}\right)$, which is defined by the following constraints.

$$z^{\mathrm{a}}_{t+1,l} = z^{\mathrm{ax}}_{tl} + z^{\mathrm{tx}}_{t,l-1,t-\Delta_{l-1}+1} - X^{\mathrm{r}}_{t+1,l} \qquad \forall t \in \mathcal{T}, l \in \mathcal{L} \tag{2.7}$$
$$z^{\mathrm{t}}_{t+1,l,t'} = z^{\mathrm{tx}}_{t,l,t'} \qquad \forall t \in \mathcal{T}, l \in \mathcal{L}_0, t - \Delta_l + 2 \leq t' \leq t \tag{2.8}$$

Constraints (2.7) track the number of available workers at each level in each time period. This number is increased at time $t + 1$ by the number of workers who started their training at $t - \Delta_l + 1$ at level $l - 1$, and decreased by those that resign. Workers who started their training later continue training as indicated by constraints (2.8). Finally, the composite transition function from pre-decision to pre-decision state is defined as $z_{t+1} = S^{\mathrm{M}}\left(z_t, x_t, X_t^{\mathrm{r}}\right) = S^{\mathrm{MX}}\left(S^{\mathrm{Mx}}\left(z_t, x_t\right), X_t^{\mathrm{r}}\right)$.

Extending the previous example and assuming that level 2 worker B resigns ($X_{1,2}^{\mathrm{r}} = 1$), the new state $z_1 = S^{\mathrm{MX}}\left(z_0^{\mathrm{x}}, X_1^{\mathrm{r}}\right)$ becomes

- worker C is available, but worker B has resigned
  ($z_{1,2}^{\mathrm{a}} = z_{0,2}^{\mathrm{ax}} + z_{0,1,-2}^{\mathrm{tx}} - X_{1,2}^{\mathrm{r}} = 1 + 1 - 1 = 1$)
- worker D is still in initial training ($z_{1,0,0}^{\mathrm{t}} = z_{0,0,0}^{\mathrm{tx}} = 1$)
- worker A is still in training from Level 1 to 2 ($z_{1,1,0}^{\mathrm{t}} = z_{0,1,0}^{\mathrm{tx}} = 1$)

2.4.3. Objective function and costs

The one-period costs are a function of the post-decision state and consist of workforce costs, i.e., salaries and training costs, and the penalty costs for under-staffing. The objective is to minimize the total discounted costs, where γ denotes the discount factor. Because the state at any future time is random, the objective is to find a policy that minimizes the expected costs over all outcomes. Let Π denote the set of policies. A *policy* $\pi \in \Pi$ specifies a sequence of state-dependent decision rules, i.e., a decision rule to be used for each state. A *stationary policy* uses the same decision rule for each decision. An *optimal policy* is defined to be one that produces an objective function value that is as good as or better than the value of any other policy for all states. Because the problem data are stationary, an optimal policy is stationary as well (see Puterman, 2014, Chapter 6). Hence, I restrict my attention to stationary policies. Let $A^{\pi}\left(z_t\right) \mapsto x_t$ denote the decision rule of stationary policy π. This leads to the objective function

$$\min_{\pi \in \Pi} \mathbb{E} \sum_{t=0}^{\infty} \gamma^t C\left(S^{\mathrm{Mx}}\left(z_t, A^{\pi}\left(z_t\right)\right)\right) \tag{2.9}$$

where $C\left(z_t^{\mathrm{x}}\right) = C^{\mathrm{wf}}\left(z_t^{\mathrm{x}}\right) + C^{-}\left(z_t^{\mathrm{x}}\right)$ is the one-period cost function of the post-decision state consisting of workforce costs $C^{\mathrm{wf}}\left(z_t^{\mathrm{x}}\right)$ and the under-staffing penalty $C^{-}\left(z_t^{\mathrm{x}}\right)$.

Workforce costs. Let c_l^{t} denote the costs for a worker in training from level $l \in \mathcal{L}_0$ to $l+1$, and let c_l^{s} denote the regular salary for a level l worker. Furthermore, let $z_{tl}^{\mathrm{tx}} \equiv \sum_{t'=\mathcal{T}_{tl} \cup \{t\}} z_{tlt'}^{\mathrm{tx}}$. The workforce costs are linear in the post-decision state; that is,

$$C^{\mathrm{wf}}\left(z_t^{\mathrm{x}}\right) = \sum_{l \in \mathcal{L}_0} c_l^{\mathrm{t}} z_{tl}^{\mathrm{tx}} + \sum_{l \in \mathcal{L}} c_l^{\mathrm{s}} z_{tl}^{\mathrm{ax}} \tag{2.10}$$

Under-staffing penalty. Experienced workers are fully productive at their current level while those in training from level $l \in \mathcal{L}_0$ to $l+1$ have reduced availability. In particular, the latter contribute to capacity at the target level $l+1$ at potentially a negative rate ρ_l^{t}, which is a function of their relative productivity at level $l+1$ and the amount of supervision required during on-the-job training. Also, they contribute to capacity at their current level l at rate ρ_l^{c}, which corresponds to their remaining capacity after on-the-job and off-the-job training is accounted for. Since a worker in training cannot be more productive than an available worker, it is assumed that $\rho_l^{\mathrm{c}} + \rho_l^{\mathrm{t}} \leq 1$. The capacity at each level $l \in \mathcal{L}$ is

$$K_l\left(z_t^{\mathrm{x}}\right) = z_{tl}^{\mathrm{ax}} + \sum_{k=l:k<L} \rho_l^{\mathrm{c}} z_{tl}^{\mathrm{tx}} + \rho_{l-1}^{\mathrm{t}} z_{t,l-1}^{\mathrm{tx}} \tag{2.11}$$

Having reached skill level $l \in \mathcal{L}$ allows a worker to satisfy demand at all levels $k \leq l$. Therefore, capacity $K_l\left(z_t^{\mathrm{x}}\right)$ can be distributed to levels $k = 1, \ldots, l$. Now let D_l denote the demand, i.e., the minimum number of required workers per period at level $l \in \mathcal{L}$, and let c^- denote the penalty cost per unit of capacity shortage per period. The total penalty costs per period are calculated as follows (see Algorithm 1). Let K denote the remaining capacity, which is zero initially. Starting at the top level down to the lowest level, the available capacity of that level is added to K. If K covers the demand, then it is reduced by the demand. If the demand is greater than K, then the penalty is increased to account for the uncovered demand and K is set to zero. Note that $K_l\left(z_t^{\mathrm{x}}\right)$ may be negative, thus decreasing K at level l.

Algorithmus 1 : Under-staffing penalty

Function $C^{-}(z_t^x) : \mathbb{R}$
  Input : Current post-decision state z_t^{x}
  Output : total under-staffing penalty per period, $C^{-}(z_t^{\mathrm{x}})$, denoted by c below

  $c \leftarrow 0$, $K \leftarrow 0$
  for $l = L, \ldots, 1$ **do**
    $K \leftarrow K + K_l(z_t^{\mathrm{x}})$
    if $K \geq D_l$ **then** $K \leftarrow K - D_l$
    else $c \leftarrow c + c^{-} \cdot (D_l - K)$, $K \leftarrow 0$
  return c

2.5. Solution Methodology

In this section, various solution methods to solve model (2.1) – (2.9) are developed. If the state space is sufficiently small, optimal solutions can be derived with policy iteration (PI) or value iteration (VI) (see Powell, 2007; Puterman, 2014). Unfortunately, the state space for instances of realistic size is far too large for the PI or VI algorithms, even if worker counts are bounded. As an alternative, I employ heuristic decision rules or *policy function approximations,* which directly return an action given a state, without resorting to any form of embedded optimization (Powell, 2007). These decision rules are aimed at maintaining certain workforce inventories or stocks. Hence, they are parameterized by *base-stock values* expressing the number of workers the employer strives to maintain at each level. The first decision rule uses only this parameter vector, and I call it *base-stock policy* or *base-stock rule.* Base-stock values and current stocks indicate whether there is a shortage or excess at each level and how many workers need to be hired and trained to compensate for potential shortages. Sending too many employees to training can create too large of a shortage at their current level. Therefore, the second decision rule uses an additional parameter vector, which determines *minimum-stock values* that are not undercut when sending workers to training. The result is a parameterized decision rule I call *base/minimum-stock policy* or *base/minimum-stock rule.* When extensive supervision is needed, i.e. when workers in training reduce the capacity at the target level, having many workers simultaneously in training creates shortages at the target level. To prevent such situations, the third policy uses a third parameter vector that specifies *training-capacity values.* The result is a policy I call *base/minimum-stock/training-capacity policy* or *base/minimum-stock/training-capacity rule*. The difficulty is finding good parameter values. For that purpose, simulation is used to obtain estimates of the value of a policy. Key to the computations is an effective search strategy to deal with noisy measurements. The notation used in this section is summarized in Table A.2 in Appendix A.

2.5.1. Policies

Let S_l denote the base-stock (desired number of workers) at level $l \in \mathcal{L}$ and let $S = (S_1, \ldots, S_L)$. Let $z_{tl} = z^{\mathrm{a}}_{tl} + \sum_{t' \in \mathcal{T}_{t,l-1}} z^{\mathrm{t}}_{t,l-1,t'}$ be the number of workers at level $l \in \mathcal{L}$ or in training to reach level l at time t and let $n_{tl} = \sum_{k=l}^{L} (S_k - z_{tl})$ be the shortage ($n_{tl} > 0$) or surplus ($n_{tl} < 0$) at level $l \in \mathcal{L}$ at time t. Note how n_{tl} accounts for downgrading by including potential excesses of workers at all levels above level l.

Definition. A *base-stock policy* specifies a decision rule $A(S, z_t) \mapsto x_t$ for all t such that

$$x_{tl} = \min\left\{[n_{t,l+1}]^+, z^{\mathrm{a}}_{tl}\right\} \qquad \text{for } l = L-1, L-2, \ldots, 1 \tag{2.12}$$

$$x_{t0} = [n_{t,1}]^+ = \left[\sum_{l \in \mathcal{L}} (S_l - z_{tl})\right]^+. \tag{2.13}$$

where $[\,\cdot\,]^+$ denotes $\max\{0, \cdot\}$.

Equation (2.12) determines the number of workers that are sent to training at all levels, excluding the hiring decision. Hence, at each level we try to send as many workers to training as necessary to compensate for the shortage at the next higher level, but no more than available. If there is a surplus at the next higher level, no workers are trained. Equation (2.13) determines the number of workers to hire. The absence of hiring restrictions means that it is always possible to directly address the demand for entry workers. Therefore, the hiring decision (x_{t0}) adjusts the total size of the workforce to the sum of the base-stock values if there are too few workers in the system. The computations are carried out incrementally starting at the top level with $n_{tL} = S_L - z_{tL}$ and $n_{tl} = n_{t,l+1} + S_l - z_{tl}$ for all lower levels.

As an example, assume a problem instance with two skill levels and let $D_1 = D_2 = 1$, $S_1 = 3$, $S_2 = 5$, $z^{\mathrm{a}}_{t1} = 4$, $z^{\mathrm{a}}_{t2} = 1$, $\rho^{\mathrm{t}}_0 = \rho^{\mathrm{c}}_1 = \rho^{\mathrm{t}}_1 = 0$. Since there is a shortage of four workers at level 2 ($n_{t2} = S_2 - z_{t2} = 4$), with the base-stock policy, all four level 1 workers will be trained immediately and three workers will be hired ($n_{t1} = 3$). This creates a manpower shortage of one at level 1, however, so it may not be the best option. Because there is an oversupply of one at level 1 ($S_1 + 1 = z^{\mathrm{a}}_{t1}$), the employer can send one worker to train with assurance that the stock does not drop below the base-stock value. Training two workers would create a temporary shortage of one worker at level 1 compared to the base-stock value, but the demand would still be satisfied, as long as the remaining two workers do not both resign. Training three workers would create a temporary shortfall of two workers at level 1 compared to the base-stock value, but the demand would still be satisfied, as long as the remaining worker does not resign.

These options raise the question: "should a second and third worker be trained immediately or is it better to wait until newly-hired recruits become level 1 workers and create an oversupply?" One extreme is to train as many workers as possible $\left(x_{tl} = \min\left\{z^{\mathrm{a}}_{tl}, [n_{t,l+1}]^{+}\right\}\right)$ as done with the base-stock policy; the other extreme is to never violate the base-stock value $\left(x_{tl} = [\min\{z^{\mathrm{a}}_{tl} - S_l, n_{t,l+1}\}]^{+}\right)$. As any value between these bounds may be suitable, I introduce a parameter vector $M = (M_1, \ldots, M_{L-1})$ of size $l - 1$, where M_l denotes the minimum-stock for $l \in \{1, \ldots, L-1\}$ that must be maintained when sending workers to training from level l to $l + 1$.

Definition. The *base/minimum-stock policy* uses the decision rule $A(S, M, z_t) \mapsto x_t$ for all t defined by

$$x_{tl} = [\min\{n_{t,l+1}, z^{\mathrm{a}}_{tl} - M_l\}]^{+} \qquad \text{for } l = L-1, L-2, \ldots, 1 \tag{2.14}$$

$$x_{t0} = [n_{t,1}]^{+}. \tag{2.15}$$

Equation (2.14) determines the number of workers who are sent to training at all levels, excluding the hiring decision. The employer tries to send as many workers to training as necessary to compensate for the shortage at the next higher level, but a minimum of M_l workers must remain at the current level $\left(x_{tl} \leq [z^{\mathrm{a}}_{tl} - M_l]^{+}\right)$. If that is not possible or if $n_{t,l+1} < 0$, no workers are trained. Equation (2.15) determines the number of workers to hire. In the previous example, if $M_1 = 2$, two workers are trained from level 1 to 2 ($n_{t2} = 4, z^{\mathrm{a}}_{tl} - M_l = 2$), leaving two workers at level 1, and three workers are hired.

As a variation of the above example, assume that each new trainee requires half-time supervision and does not add to capacity. Accordingly, $\rho_0^{\mathrm{t}} = -\frac{1}{2}$. Given that three workers are hired, the capacity at level 1 is temporarily reduced to $K_1 = 2 - 1.5 = 0.5$ creating a manpower shortage of 0.5. In such a situation, it may be better to train two workers at a time. Then, if no level 1 worker resigns, the capacity remains at $K_1 = 2 - 1 = 1$, which is sufficient to cover the demand. To deal with this case, an additional parameter is used for each level l where $\rho_l^{\mathrm{t}} < 0$. Specifically, let C_l denote the maximum number of level l workers who are allowed to be in training simultaneously for promotion to level $l+1$, for $l \in \{l \in \mathcal{L}_0 \mid \rho_l^{\mathrm{t}} < 0\}$. Henceforth, $C = (C_l)_{l \in \{l \in \mathcal{L}_0 \mid \rho_l^{\mathrm{t}} < 0\}}$ is referred to as the *training-capacity.*

Definition. The *base/minimum-stock/training-capacity policy* uses the decision rule $A(S, M, C, z_t) \mapsto x_t$ for all t defined by

$$x_{tl} = \begin{cases} \left[\min\left\{n_{t,l+1}, z_{tl}^{\mathrm{a}} - M_l, C_l - \sum_{t' \in \mathcal{T}_{tl}} z_{tlt'}^{\mathrm{t}}\right\}\right]^+ & \text{if } \rho_l^{\mathrm{t}} < 0 \\ \left[\min\left\{n_{t,l+1}, z_{tl}^{\mathrm{a}} - M_l\right\}\right]^+ & \text{otherwise} \end{cases} \quad \text{for } l = L-1, L-2, \ldots, 1 \tag{2.16}$$

$$x_{t0} = \begin{cases} \left[\min\left\{n_{t,1}, C_0 - \sum_{t' \in \mathcal{T}_{t0}} z_{t0t'}^{\mathrm{t}}\right\}\right]^+ & \text{if } \rho_0^{\mathrm{t}} < 0 \\ \left[n_{t,1}\right]^+ & \text{otherwise} \end{cases}. \tag{2.17}$$

Equation (2.16) determines the number of workers to train at each level, excluding the hiring decision. If the productivity rate at the target level is greater than or equal to zero, the same logic as in the base/minimum-stock policy is used. Otherwise, the number of workers who can be sent to training is limited, such that at most C_l workers can be training concurrently from level l to $l+1$. Equation (2.17) determines the number of workers to hire. Again, if the productivity rate at the target level is negative, then there can be at most C_0 workers in training.

2.5.2. Policy search

The challenge is to find the optimal base-stock, minimum-stock and training-capacity values with respect to objective function (2.9). In the trivial case when there are no resignations, i.e., $p_l^{\mathrm{r}} = 0$ for all $l \in \mathcal{L}$, the base-stock values are equal to the demand at their respective level because the state with $z^{\mathrm{a}}_{tl} = D_l$ and all other variables equal to zero must be reached once. Without further action the workforce would stay in that state forever covering demand at minimum cost. A positive resignation rate requires having a sufficient number of workers on the roster ready to fill in for the resigning workers until they can be replaced. Accordingly, the base-stock values increase with both the probabilities p^{r}_l and the training durations at level $l \in \mathcal{L}$.

As an example, Figure 2.3 illustrates how resignation probabilities may influence the optimal base-stock values of the base-stock policy for an instance with two skill levels. The gray line is the demand at each skill level $(D_1 = D_2 = 5)$. In the left graph, the resignation probability at level 2 is fixed to $p^{\mathrm{r}}_2 = 0.15$, while p^{r}_1 is varied from 0 to 0.2. An increase of p^{r}_1 is compensated for by increasing the base-stock value at level 1 (S_1). In the right graph, the resignation probability at level 1 is fixed to $p^{\mathrm{r}}_1 = 0.15$, while p^{r}_2 is varied. An increase of p^{r}_2 is compensated for by increasing the base-stock value at both levels. From $p^{\mathrm{r}}_2 = 0$ to $p^{\mathrm{r}}_2 = 0.01$, the sum of the base-stock values remains the same, but the number of level 1 workers decreases by one and the number of level 2 workers, who can cover some of the demand at level 1 due to the possibility of downgrading, increases by one.

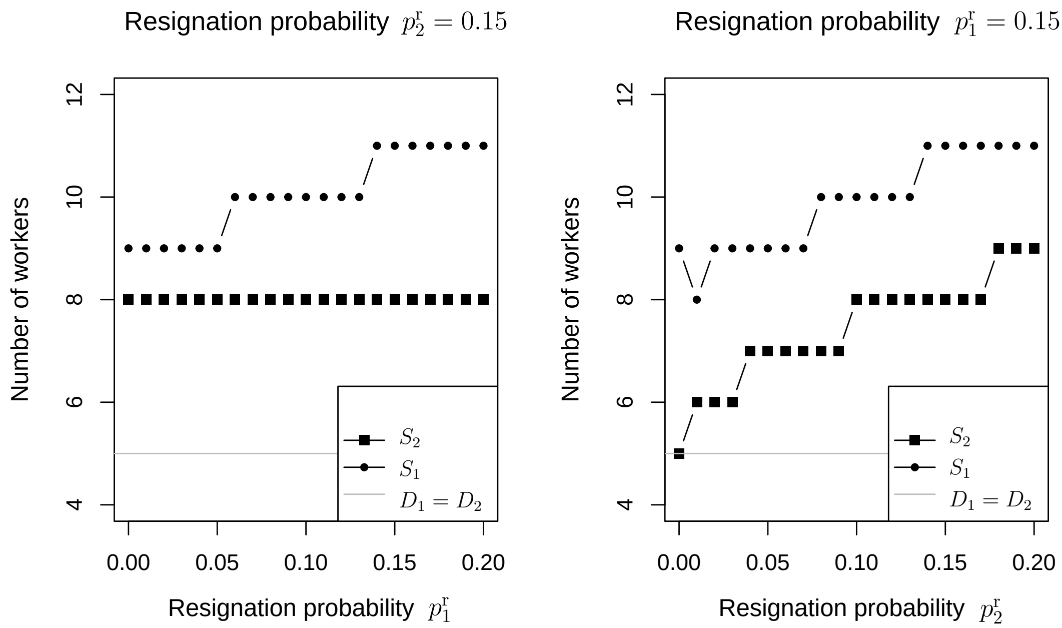


Figure 2.3.: Base-stock values as a function of resignation probabilities

Policy evaluation. In the search for good parameter values it is necessary to evaluate policies. If the state space is sufficiently small, it is possible to calculate the exact value of a policy for each state as in the PI algorithm. However, for instances reflecting reality in the targeted industries, even a finite state space would be much too large for the policy evaluation step in PI to succeed. As an alternative, the value of a policy is statistically estimated via simulation. There are two issues to consider. First, the objective function value for a policy depends on the initial state. Since the policy function approximations $A(S, z_t)$, $A(S, M, z_t)$, and $A(S, M, C, z_t)$ are not optimal, they may perform well in general, but lead to inferior decisions in some states. Second, for infinite horizon problems it is impossible to obtain an unbiased observation of the objective function value with a finite number of measurements (Powell, 2007, Section 9.1.2).

To address the second issue, instead of directly simulating the objective function value, which would introduce a bias, the expected one-period costs are estimated and then projected out over an infinite horizon. The error in the objective function value is then solely a result of the deviation between the true and the estimated expected one-period costs.

To address the first issue, observe that the aim is to determine a “good” policy, that is, one that performs well for the states that are visited most often under that policy. Therefore, the simulation initially starts in the state with $z^{\mathrm{a}}_{0l} = S_l$ and $z^{\mathrm{t}}_{0lt'} = 0$, and all subsequent simulation runs continue from the state where the previous simulation run ended. Hence, the more periods are simulated, the smaller the effect of the initial state gets, and the greater the accuracy of the average one-period costs under that policy gets. In other words, the average costs are weighted according to the frequency in which states are visited using that policy.

Simulating a single period given policy π and state z_t at time t requires the following operations: (i) determine the post decision state $\left(z^{\mathrm{x}}_t \leftarrow S^{\mathrm{Mx}}\left(z_t, A^{\pi}\left(z_t\right)\right)\right)$; (ii) compute the one-period costs, which are denoted by $\hat{C}^{\pi}_t \leftarrow C\left(z^{\mathrm{x}}_t\right)$; (iii) obtain a sample of the resignations, which is denoted by $\hat{X}^{\mathrm{r}}_{t+1}$; and (iv) determine the next state that can be used as input for simulating another period (see Algorithm 2). Let $\bar{C}^{\pi}_n = \frac{1}{n}\sum_{t=0}^{n-1}\hat{C}^{\pi}_t$ denote the sample mean after n observations. Using the discount factor γ, the objective function value (2.9) is estimated as $\bar{C}^{\pi}_n / \left(1-\gamma\right)$.

Algorithmus 2 : Policy evaluation simulation

Function *sim*
  Input : Policy π specifying a decision rule $A^{\pi} : \mathcal{Z} \rightarrow \mathcal{X}$,
  State z_t
  Output : Computed one-period costs $\hat{C}^{\pi}_t$,
  Next state $\hat{z}_{t+1}$

  $z^{\mathrm{x}}_t \leftarrow S^{\mathrm{Mx}}\left(z_t, A^{\pi}\left(z_t\right)\right)$
  $\hat{C}^{\pi}_t \leftarrow C\left(z^{\mathrm{x}}_t\right)$
  $\hat{X}^{\mathrm{r}}_{t+1} \leftarrow$ sample X^{r}_{t+1}
  $\hat{z}_{t+1} \leftarrow S^{\mathrm{MX}}\left(z^{\mathrm{x}}_t, \hat{X}^{\mathrm{r}}_{t+1}\right)$
  return $\hat{C}^{\pi}_t$ and $\hat{z}_{t+1}$

State of knowledge. For comparing two policies, the sample size, the sample mean and the sample variance are required as described below. Let n^{π}, $M^{2,\pi}_n = \sum_{i=1}^{n}\left(\hat{C}^{\pi}_i - \bar{C}^{\pi}_n\right)^2$, $S^{2,\pi}_n = \frac{1}{n-1}M^{2,\pi}_n$, and $S^{\pi}_n = \sqrt{S^{2,\pi}_n}$ denote the sample size, the sum of the squares of the differences from the sample mean, the sample variance, and the sample standard deviation after n observations, respectively. These quantities can be updated recursively as follows (see Welford, 1962).

$$\bar{C}_n^{\pi} = \left(1 - \frac{1}{n}\right) \bar{C}_{n-1}^{\pi} + \frac{1}{n}\hat{C}_n^{\pi} \tag{2.18}$$

$$M_n^{2,\pi} = M_{n-1}^{2,\pi} + \frac{n-1}{n}\left(\hat{C}_n^{\pi} - \bar{C}_{n-1}^{\pi}\right)^2 \tag{2.19}$$

To improve efficiency, the policy parameters and the triplets n^{π}, $\bar{C}_n^{\pi}$ and $M_n^{2,\pi}$ are stored in a hash map V for constant time access. Each time a policy π is evaluated for the first time, its triplet is initialized with $n^{\pi} = 1$, $\bar{C}_1^{\pi} = \hat{C}_1^{\pi}$, $M_1^{2,\pi} = 0$ and placed in V. If a triplet for π already exists in V, it is updated with the recursive equations (2.18) and (2.19).

Search strategy. Finding good policy parameters via simulation is challenging because (i) the parameter vectors are integer points in a multi-dimensional space, and hence, there are many candidate vectors to evaluate, and (ii) policy evaluation is based on noisy measurements that are computationally expensive. Reducing the noise requires many observations. A naive procedure would be to conduct a neighborhood search where each policy is simulated for a sufficient number of periods. However, it is more efficient to avoid excess computations on candidates with little promise, but instead invest in determining the best candidate out of a few promising alternatives. Let $N(\pi)$ denote the neighborhood of policy π. The neighborhoods for the base-stock, the base/minimum-stock and the base/minimum-stock/training-capacity policy are defined as follows.

$$\begin{aligned}
N(S) &= \left\{S' \mid [S_l - 1]^+ \leq S'_l \leq S_l + 1, S'_L \geq D_L\right\} \setminus S \\
N(S, M) &= \left\{S' \times M' \mid S' \in N(S), [M_l - 1]^+ \leq M'_l \leq M_l + 1, M'_l \leq S'_l\right\} \setminus (S, M) \\
N(S, M, C) &= \Bigg\{S' \times M' \times C' \mid (S', M') \in N(S, M), \max\{C_l - 1, 1\} \leq C'_l \leq \\
&\qquad \min\left\{C_l + 1, \sum_{k=l+1}^{L} S'_k\right\}\Bigg\} \setminus (S, M, C),
\end{aligned}$$

For the current policy π, the basic logic of the search is as follows:

1. terminate when there is no better neighbor as determined statistically, or

2. move to a better neighbor when a better neighbor is found, or

3. perform additional measurements for either π or a promising $\phi \in N(\pi)$ to increase the sample size, and then continue.

The search terminates at a local optimum π when all neighbors $\phi \in N(\pi)$ do not provide a better objective function value than π. To statistically decide if policy π is better than a policy ϕ, i.e., if $\bar{C}^{\pi}_{\infty} < \bar{C}^{\phi}_{\infty}$ we use Welch's t-test (see e.g., Ruxton, 2006). The test statistic normalizes the mean difference with the approximate standard deviation of the mean difference:

$$T(\pi, \phi) = \frac{\bar{C}^{\pi}_{n^{\pi}} - \bar{C}^{\phi}_{n^{\phi}}}{\sqrt{\frac{S^{2,\pi}_{n^{\pi}}}{n^{\pi}} + \frac{S^{2,\phi}_{n^{\phi}}}{n^{\phi}}}}. \tag{2.20}$$

The null hypothesis is $\bar{C}^{\pi}_{\infty} \geq \bar{C}^{\phi}_{\infty}$ and the rejection region given the significance level α is defined by $\{T(\pi, \phi) \mid T(\pi, \phi) < -t_{1-\alpha;\nu}\}$, where $t_{1-\alpha;\nu}$ is the $(1-\alpha)$-quantile with ν degrees of freedom of the Student's t-distribution, and ν is calculated as

$$\nu \approx \frac{\left(\frac{S^{2,\pi}_{n^{\pi}}}{n^{\pi}} + \frac{S^{2,\phi}_{n^{\phi}}}{n^{\phi}}\right)^2}{\frac{\left(\frac{S^{2,\pi}_{n^{\pi}}}{n^{\pi}}\right)^2}{n^{\pi}-1} + \frac{\left(\frac{S^{2,\phi}_{n^{\phi}}}{n^{\phi}}\right)^2}{n^{\phi}-1}}. \tag{2.21}$$

The difficulty is that when $\bar{C}^{\pi}_{\infty}$ and $\bar{C}^{\phi}_{\infty}$ are almost equal, a vast number of observations is needed to reject the null hypothesis given a fixed significance level. To limit the computation time, the number of observations that are collected per pair of policies π and ϕ is limited with parameter *obs*. The significance level is

$$\alpha(\pi, \phi) = 0.0001 + \frac{0.5 - 0.0001}{obs} \cdot \min\left\{n^{\pi}, n^{\phi}\right\}, \tag{2.22}$$

which is gradually increased from 0.0001 to 0.5. When $\min\left\{n^{\pi}, n^{\phi}\right\} = obs$, $\alpha(\pi, \phi) = 0.5$ and the Welch-test is satisfied when $T(\pi, \phi) < 0$. Otherwise, if at any time $T(\pi, \phi) \geq 0$, the search moves to policy ϕ. The search procedure is outlined in Algorithm 3.

Algorithmus 3 : Policy search

Input : Initial policy π
Output : New policy π

$obs \leftarrow 12.5 \cdot 10^6$
Initialize hashmap V
mainloop: **while** *true* **do**
    Observe π, update $V(\pi)$
    for $\phi \in N(\pi)$ **do**
        Observe ϕ, update $V(\phi)$
        innerloop: **while** *true* **do**
            if $T(\pi, \phi) > 0$ **then** $\pi \leftarrow \phi$, **continue** mainloop
            if $T(\pi, \phi) < -t_{1-\alpha(\pi,\phi);\nu}$ **then** **break** innerloop
            if $n_\pi < n_\phi$ **then** Observe π, update $V(\pi)$ **else** Observe ϕ, update $V(\phi)$
    return π

Computational complexity. Simulating a single period (Algorithm 2) requires the following: evaluation of the decision rule, transition to the post-decision state $\left[S^{\mathrm{Mx}}(z_t, x_t)\right]$, calculation of the costs $C(z_t^{\mathrm{x}})$, sampling to determine the number of resignations, and transition to the next pre-decision state $\left[S^{\mathrm{MX}}\left(z_t^x, \hat{X}_{t+1}\right)\right]$. For simplicity, let $N = \max_{l \in \mathcal{L}_0} \{\Delta_l, L\}$. Evaluating the decision rule has a complexity $\mathcal{O}(N^2)$ because it has to loop over the training duration of each skill level to determine the number of workers that will be at each level. The cost function has time complexity $\mathcal{O}(N^2)$ because all state variables may contribute to the costs and the capacities. The complexity of the transition functions is in $\mathcal{O}(N^2)$ because all state variables must be set. Therefore, simulating one period has a complexity of $\mathcal{O}(N^2)$.

When comparing two policies, at most $2 \cdot obs$ simulation runs are needed to decide which is better. In each iteration of the "mainloop" of Algorithm 3 the current policy π is compared to at most $|N(\pi)|$ neighbors. Because $|N(S)| \approx \mathcal{O}\left(3^L\right)$, $|N(S, M)| \approx \mathcal{O}\left(3^{2L}\right)$ and $|N(S, M, C)| \approx \mathcal{O}\left(3^{3L}\right)$ the neighborhoods grow exponentially in L, and thus the "mainloop" has exponential complexity that depends on the decision rule. The number of iterations depends on where the search starts and the final policy parameters. For the evaluated data sets, I found that less than 100 iterations were required.

2.6. Computational Study

In the first set of experiments, I demonstrate the effectiveness of the base-stock and base/minimum-stock policies in combination with the proposed search procedure to generate solutions for small to large instances. Whenever possible, the results of the presented procedure are compared to the results obtained with the standard algorithms PI and VI (see Powell, 2007; Puterman, 2014). To determine the value of downgrading, each instance is solved another time with a modified cost function that does not allow for downgrading. Next, I examine the performance of the base/minimum-stock/training-capacity policy when on-the-job training requires significant supervision (significantly reduces capacity at the target levels) and compare the results to the base/minimum-stock policy. The results show that the base/minimum-stock/training-capacity policy performs significantly better than the base/minimum-stock policy, but the computational effort is high. Finally, I show that short and intense training periods are superior to long training periods with low intensity. In the experimental design, all costs are given in 1000 monetary units and the discounting factor γ is set to 0.99. A 12-hour time limit was placed on the computations for each instance, which were conducted on a PC with a quad core Intel®Xeon®CPU E3-1225 v3 running at a frequency of 3.20GHz, and with 16 GB of storage.

2.6.1. Base-stock and base/minimum-stock policies

In the first part of the analysis, the ability of the base-stock and base/minimum-stock policies in combination with the search procedure to solve instances of increasing size is demonstrated. Extending the number of skill levels and training durations adds dimensions to the state space. Higher demand and resignation rates require maintaining a larger workforce. The number of skill levels L is limited to 2, 3 and 4. To make the number of test instances manageable, specific model parameters are fixed based on steel and chemical industry norms and worst case assumptions. The productivity rates are non-negative except for two instances. I increase the size on an instance by increasing the training durations, the demand, and the resignation probabilities. The penalty cost per unit of capacity shortage per period is set to $c^- = 1000$ for all instances. Table 2.2 lists the 32 instances that were generated for demonstration purposes. While instances 1 to 30 have been generated by systematically ranging the problem parameters, instances 31 and 32 are based on real world data sets. Workforce demand, salaries, training durations and costs, as well as productivity and resignation rates are stated per period.

The results for PI, VI, the base-stock and the base/minimum-stock policy are presented in Table 2.3. Computation times (sec) and performance measures are given for each algorithm. When an algorithm fails to solve an instance due to an out-of-memory error, it is indicated by "OOM"; when an algorithm has not terminated within 12 hours, it is indicated by "> 12h". For PI and VI the table shows the average cost per period for an optimal policy when the system starts in the best state $C^* = (1-\gamma) \cdot \min_{z_t \in \mathcal{Z}} V(z_t)$, where $V(z_t)$ is the value of the objective function (2.9). For the base-stock and the base/minimum-stock policies the final policies are simulated another 10^7 periods to obtain a reliable estimate for the average cost per period (column $\bar{C}^\pi$). The column "Gap to PI, VI" gives the gap with respect to an optimal policy, if available, where Gap $= \left(\bar{C}^\pi - C^*\right)/C^*$. Furthermore, in order to determine the value of downgrading, the instances have additionally been solved with the base/minimum-stock policy and a modified cost function that does not account for downgrading. More precisely, the original under-staffing penalty $C^-(z_t^{\mathrm{x}})$ as shown in Algorithm 1, is replaced by $C^-(z_t^{\mathrm{x}}) = c^- \cdot \sum_{l \in \mathcal{L}} [D_l - K_l(z_t^{\mathrm{x}})]^+$. Again, the average costs are shown (column $\bar{C}^{\pi\prime}$), and the column "Gap to A^{SM}" reports the gap with respect to the base/minimum-stock policy with downgrading.

The results show that the standard algorithms are only capable of solving small instances. For instances of reasonable size, PI runs out of memory (OOM) at the policy evaluation step due to the required matrix inversion (see Powell, 2007, Section 3.3 Equation (3.20)); for VI the computation time becomes prohibitively long. In contrast, with the proposed heuristic, solutions are found for all instances. The longest runtime of 223 minutes was observed for the base/minimum-stock policy for instance 31. In terms of solution quality, the base-stock policy exhibits large gaps of up to 18.5% and 19.6% with respect to the available optimal results and the base/minimum-stock policy, respectively. The gaps indicate that, in general, the base-stock policy performs better when workers are productive during training, i.e., when ρ_l^{c} are comparatively high. For example, instances 1 and 2 are identical, except for the productivity rate ρ_1^{c}, which is 0.3 for instance 1 and 0.0 for instance 2. The optimality gap for instance 2 (0.5) is larger than for instance 1 (0.3). This can be observed for all such pairs [i.e., all instance pairs $(1,2), (3,4), \ldots, (29,30)$] where all problem parameters are identical except for the ρ_l^{c} values, by either looking at the optimality gap (column "Gap to PI, VI") if available or by looking at the gap with respect to the

Instance	No. skill levels L	Min no. workers per level D_l	Salary per level c_l^{s}	Training				Resignation probability per level p_l^{r}
				Duration per level Δ_l	Costs per level c_l^{t}	Productivity starting level per level, ρ_l^{c}	Productivity target level per level, ρ_l^{t}	
1	2	$1, 1$	$10, 20$	$1, 1$	$5, 15$	0.3	$0, 0$	$0.02, 0.01$
2		$1, 1$		$1, 1$		0.0		$0.02, 0.01$
3		$2, 2$		$2, 2$		0.3		$0.04, 0.02$
4		$2, 2$		$2, 2$		0.0		$0.04, 0.02$
5		$3, 3$		$3, 3$		0.3		$0.05, 0.02$
6		$3, 3$		$3, 3$		0.0		$0.05, 0.02$
7		$7, 7$		$4, 4$		0.3		$0.06, 0.03$
8		$7, 7$		$4, 4$		0.0		$0.06, 0.03$
9		$15, 15$		$7, 7$		0.3		$0.08, 0.04$
10		$15, 15$		$7, 7$		0.0		$0.08, 0.04$
11	3	$1, 1, 1$	$20, 26, 32$	$1, 1, 1$	$6, 21, 26$	$0.3, 0.3$	$0, 0, 0$	$0.03, 0.02, 0.01$
12		$1, 1, 1$		$1, 1, 1$		$0.0, 0.0$		$0.03, 0.02, 0.01$
13		$2, 2, 2$		$2, 2, 2$		$0.3, 0.3$		$0.04, 0.03, 0.02$
14		$2, 2, 2$		$2, 2, 2$		$0.0, 0.0$		$0.04, 0.03, 0.02$
15		$3, 3, 3$		$3, 3, 3$		$0.3, 0.3$		$0.05, 0.04, 0.03$
16		$3, 3, 3$		$3, 3, 3$		$0.0, 0.0$		$0.05, 0.04, 0.03$
17		$7, 7, 7$		$4, 4, 4$		$0.3, 0.3$		$0.07, 0.04, 0.03$
18		$7, 7, 7$		$4, 4, 4$		$0.0, 0.0$		$0.07, 0.04, 0.03$
19		$15, 15, 15$		$7, 7, 7$		$0.3, 0.3$		$0.1, 0.07, 0.04$
20		$15, 15, 15$		$7, 7, 7$		$0.0, 0.0$		$0.1, 0.07, 0.04$
21	4	$1, 1, 1, 1$	$20, 30, 40, 50$	$1, 1, 1, 1$	$6, 20, 30, 40$	$0.3, 0.3, 0.3$	$0, 0, 0, 0$	$0.04, 0.03, 0.02, 0.01$
22		$1, 1, 1, 1$		$1, 1, 1, 1$		$0.0, 0.0, 0.0$		$0.04, 0.03, 0.02, 0.01$
23		$2, 2, 2, 2$		$2, 2, 2, 2$		$0.3, 0.3, 0.3$		$0.05, 0.04, 0.03, 0.02$
24		$2, 2, 2, 2$		$2, 2, 2, 2$		$0.0, 0.0, 0.0$		$0.05, 0.04, 0.03, 0.02$
25		$3, 3, 3, 3$		$3, 3, 3, 3$		$0.3, 0.3, 0.3$		$0.06, 0.05, 0.04, 0.03$
26		$3, 3, 3, 3$		$3, 3, 3, 3$		$0.0, 0.0, 0.0$		$0.06, 0.05, 0.04, 0.03$
27		$7, 7, 7, 7$		$4, 4, 4, 4$		$0.3, 0.3, 0.3$		$0.08, 0.06, 0.04, 0.03$
28		$7, 7, 7, 7$		$4, 4, 4, 4$		$0.0, 0.0, 0.0$		$0.08, 0.06, 0.04, 0.03$
29		$15, 15, 15, 15$		$7, 7, 7, 7$		$0.3, 0.3, 0.3$		$0.1, 0.06, 0.05, 0.04$
30		$15, 15, 15, 15$		$7, 7, 7, 7$		$0.0, 0.0, 0.0$		$0.1, 0.06, 0.05, 0.04$
31	3	$52, 8, 12$	$20, 26, 32$	$7, 6, 2$	$6, 21, 26$	$0.2, 0.68$	$0.0, -0.07, 0.2$	$0.1, 0.04, 0.01$
32	4	$4, 4, 8, 24$	$25, 40, 45, 55$	$7, 5, 4, 1$	$6, 25, 40, 45$	$0.2, 0.5, 0.0$	$0, -0.1, -0.1, 0$	$0.1, 0.05, 0.02, 0.01$

Table 2.2.: Test instances

Instance	Policy iter. (PI) Run-time (sec)	Policy iter. (PI) Avg. cost per period C^*	Value iter. (VI) Run-time (sec)	Value iter. (VI) Avg. cost per period C^*	Base-stock policy Run-time (sec)	Base-stock policy Avg. cost per period $\bar{C}^{\pi}$	Base-stock policy Gap to PI, VI (%)	Base-stock policy Gap to A^{SM} (%)	Base/minimum-stock pol. Run-time (sec)	Base/minimum-stock pol. Avg. cost per period $\bar{C}^{\pi}$	Base/minimum-stock pol. Gap to PI, VI (%)	Without downgr. Avg. cost per period $\bar{C}^{\pi'}$	Without downgr. Gap to A^{SM} (%)
1	< 1	50	5	50	< 1	50	0.3	0.1	4	50	0.2	50	0.0
2	< 1	50	4	50	< 1	50	0.5	0.2	6	50	0.3	50	0.0
3	24	98	172	98	1	104	6.6	6.0	1	98	0.5	104	5.7
4	19	99	112	99	< 1	106	6.8	6.4	1	99	0.4	104	4.9
5	OOM		16399	145	1	156	7.9	6.7	1	147	1.2	158	7.9
6	OOM		23852	146	4	161	10.3	8.2	1	149	1.9	159	7.1
7	OOM		> 12 h		4	337	-	3.9	4	324	-	340	5.1
8	OOM		> 12 h		3	348	-	5.5	10	330	-	344	4.4
9	OOM		> 12 h		20	763	-	2.9	21	741	-	771	4.0
10	OOM		> 12 h		19	804	-	4.9	30	767	-	792	3.4
11	15	111	981	111	1	122	9.9	9.1	2	112	0.7	134	19.2
12	2	112	498	112	1	125	11.7	10.9	2	112	0.7	134	19.1
13	OOM		> 12 h		2	255	-	9.4	5	233	-	264	13.3
14	OOM		> 12 h		2	269	-	13.9	6	236	-	266	12.5
15	OOM		> 12 h		8	400	-	9.0	15	367	-	415	13.1
16	OOM		> 12 h		7	435	-	16.4	19	374	-	425	13.7
17	OOM		> 12 h		25	853	-	8.1	138	790	-	866	9.7
18	OOM		> 12 h		36	926	-	12.9	93	820	-	892	8.8
19	OOM		> 12 h		102	2021	-	7.2	1658	1885	-	2029	7.7
20	OOM		> 12 h		177	2294	-	10.9	591	2069	-	2189	5.8
21	7845	196	> 12 h		4	227	15.6	13.8	21	199	1.6	228	14.4
22	2810	197	> 12 h		3	233	18.5	16.7	14	200	1.6	230	15.2
23	OOM		> 12 h		30	479	-	14.3	51	419	-	473	12.7
24	OOM		> 12 h		19	500	-	17.9	88	424	-	476	12.2
25	OOM		> 12 h		65	756	-	13.7	201	665	-	744	12.0
26	OOM		> 12 h		147	811	-	19.6	467	678	-	754	11.3
27	OOM		> 12 h		342	1613	-	11.2	2448	1450	-	1581	9.0
28	OOM		> 12 h		342	1725	-	14.7	723	1504	-	1624	8.0
29	OOM		> 12 h		1039	3751	-	8.4	2421	3462	-	3631	4.9
30	OOM		> 12 h		1639	4141	-	10.1	5122	3762	-	3857	2.5
31	OOM		> 12 h		191	2271	-	1.3	13355	2243	-	2374	5.8
32	OOM		> 12 h		117	2464	-	2.6	1940	2402	-	2576	7.2

Table 2.3.: Computational performance for proposed decision rules and standard algorithms

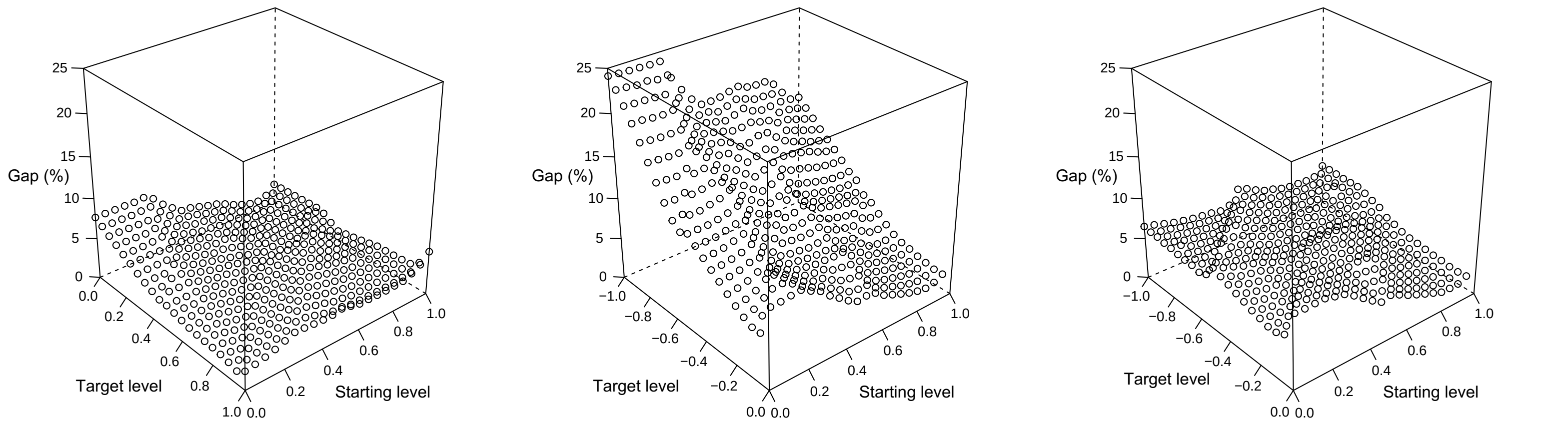
base/minimum-stock policy (column “Gap to A^{SM}“). For the base/minimum-stock policy the largest optimality gap (column “Gap to PI, VI”) we observed is 1.9% for instance 6. The increase in cost without downgrading is on average 8.8% and at most 19.2% for instance 11. These statistics highlight the value of downgrading.

2.6.2. Training productivities

The following experiment compares the solution qualities of the base/minimum-stock and the base/minimum-stock/training-capacity policies when training productivities (ρ^{c}, ρ^{t}) are incrementally varied between their extreme values. I use instances with two skill levels, training durations $\Delta_0 = \Delta_1 = 2$, minimum number of workers $D_1 = D_2 = 2$, salaries and training costs $c_0^{\mathrm{t}} = c_1^{\mathrm{s}} = c_1^{\mathrm{t}} = c_2^{\mathrm{s}} = 1$, penalty per unit of missing capacity per period $c^- = 10$, productivity at level 1 for an employee in initial training $\rho_0^{\mathrm{t}} = 0$, and resignation probabilities per period $p_1^{\mathrm{r}} = p_2^{\mathrm{r}} = 0.1$.

In the experiment, the productivities of workers being trained from level 1 to 2 are varied, and the problems are solved with the base/minimum-stock, the base/minimum-stock/training-capacity policy, and PI. Then the optimality gaps between the results provided by the heuristic policies and the optimal policies derived with PI are examined. The productivity at the starting level ρ_1^{c} is decreased in 5% steps from 100% to zero, which corresponds to reducing the time a worker has left during training. Likewise, the productivity at the target level ρ_1^{t} is decreased in 5% steps from 100% to −100%, which corresponds to less productivity and/or an increased need for supervision. All combinations of ρ_1^{c} and ρ_1^{t} values are evaluated.

For $\rho_1^{\mathrm{t}} \geq 0$ the base/minimum-stock and base/minimum-stock/training-capacity policies are equivalent. Figure 2.4a illustrates the optimality gaps of the base/minimum-stock policy for non-negative $\rho_1^{\mathrm{t}} \geq 0$. The largest gaps of up to 7.7% are observed when ρ_1^{t} and ρ_1^{c} are both close to zero; the average gap is 3.0%. Figure 2.4b illustrates the optimality gaps for the base/minimum-stock policy when $\rho_1^{\mathrm{t}} < 0$. The first observation is that there is only a weak dependence on the productivity at the starting level ρ_1^{c}; the second observation is that the gap increases (solution quality decreases) as the target level productivity decreases. Once ρ_1^{t} is −0.35 or lower, optimality gaps of 10% and greater are seen. Figure 2.4c shows the results with the base/minimum-stock/training-capacity policy for the same instances. The optimality gaps are below 7.8%, and 4.8% on average. Consequently, when on-the-job training significantly reduces capacity, the base/minimum-stock/training-capacity policy is superior to the base/minimum-stock policy.



(a) Base/minimum-stock, $\rho_1^{\mathrm{t}} \geq 0$

(b) Base/minimum-stock, $\rho_1^{\mathrm{t}} < 0$

(c) Base/min-stock/training-capacity, $\rho_1^{\mathrm{t}} < 0$

Figure 2.4.: Varying productivities during training

2.6.3. Training schedules

In order to analyze the effects of different training schedules from a management perspective, the workload for training is fixed, while the training duration is varied. When the workload is fixed and the training duration is stretched, the time spent in training per period decreases, and thus an employee in training has more time left to be productive at his current level. In addition, he spends less time on on-the-job training per period, and hence, he contributes less capacity at the target level (when capacity is contributed) and needs less supervision (when supervision is required). For example, assume that to advance to level 2, a worker needs to complete one period of on-the-job training. Therefore, the minimum possible training duration is one period if the training is conducted full time. If the training duration is stretched to four periods, a worker spends only a quarter of his time in training in each of the four periods. As a result, the productivity rate at the current level becomes $\rho_1^{\mathrm{c}} = 0.75$ and the productivity rate at the target level is quartered $\rho_1^{\mathrm{t}} \leftarrow \rho_1^{\mathrm{t}}/4$.

In the experiments I use instances with two levels and stretch the training duration from level 1 to 2 from the shortest possible time of one period up to 7 periods. To limit the number of instances, demand is fixed such that $D_1 = D_2 = 2$ per level, salaries are set to $c_0^{\mathrm{t}} = 5$, $c_1^{\mathrm{s}} = 10$, $c_1^{\mathrm{t}} = 10$, $c_2^{\mathrm{s}} = 20$, the resignation probabilities are fixed to $p_1^{\mathrm{r}} = 0.1$ and $p_2^{\mathrm{r}} = 0.08$, and the penalty costs are set at $c^{-} = 200$. Now assume that one period of on-the-job training is required to advance from level 1 to 2. Therefore, training can be completed in one period if conducted full time. Next, the productivity rate at the target level ρ_1^{t} given that training is full time (i.e., $\Delta_1 = 1$) is set to the values $-1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1$. By stretching the training duration from $\Delta_1 = 1$ to $\Delta_1 = 7$ for each value of these values, we obtain 9 sets of 7 instances each. For example, Table 2.4 shows the training parameters for the set of instances where $\rho_1^{\mathrm{t}} = -1$ given that training is full time.

Instance	1	2	3	4	5	6	7
Δ_1	1	2	3	4	5	6	7
ρ_1^{c}	0	0.5	0.67	0.75	0.8	0.83	0.86
ρ_1^{t}	-1	-0.5	-0.33	-0.25	-0.2	-0.17	-0.14

Table 2.4.: Instance set for $\rho_1^{\mathrm{t}} = -1$ for $\Delta_1 = 1$

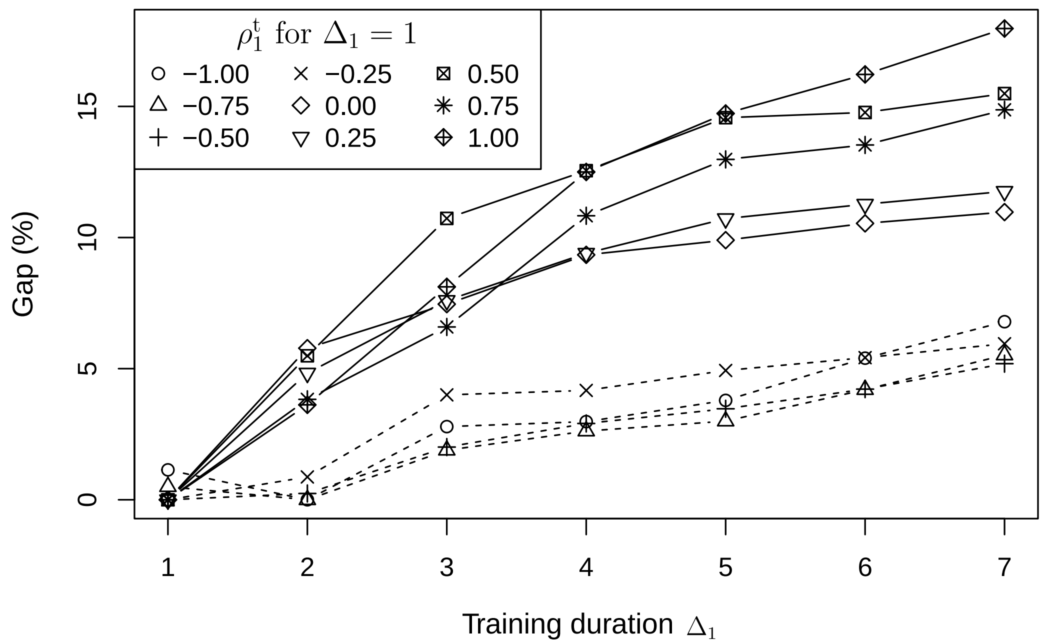


Figure 2.5.: Stretching the training duration

Each instance examined is small enough to be solved optimally with VI. The results are illustrated in Figure 2.5. Each line corresponds to one set of instances with identical problem parameters, but training durations Δ_1 varying from 1 to 7. The vertical axis gives the gap as percent increase in the objective value compared to the best objective value over all instances in the same set. Therefore, for each line, the training duration Δ_1 with gap $= 0$ gives the training duration associated with the minimum cost for all other parameters fixed.

The results show that for all sets of instances with non-negative contribution at the target level ($\rho_1^t \geq 0$) (solid lines), the instance with the shortest possible training duration (one period) yields the smallest objective function value. For some of the sets with negative contribution at the target level (dashed lines), though, the minimum objective function value is obtained when the duration is stretched to two periods. These are the instances with the lowest contributions, i.e., $\rho_1^t = -1$ and -0.75 given $\Delta_1 = 1$.

Hence, when training requires extensive supervision, employers should consider part-time training; otherwise, it is best to fully release the workers from their regular duties to allow them to complete training as quickly as possible.

2.7. Conclusions

In this chapter, an MDP model for workforce capacity planning with a hierarchically skilled workforce and random resignations was presented. Solutions to the model provide guidance on the number of employees to hire and train in light of this randomness. The objective is to balance workforce costs with the costs resulting from staff shortages. The model is novel in that employees are trained over a non-interruptible extended period of time during which their productivity declines in proportion to the intensity of the training.

To find solutions, three heuristic decision rules are presented along with a search procedure to determine the values of the accompanying parameters, i.e., desired number of workers, minimum number of workers, and training capacity for each level, respectively. Using this combined methodology I was able to solve considerably larger instances than with the standard algorithms PI and VI, which failed on instances of moderate size. In comparison, with the base/minimum-stock policy it was possible to solve realistic instances in less than four hours. Once the policy parameter values are known, decision-making is straightforward – a critical benefit from a management perspective.

Furthermore, the computations indicate that making decisions solely based on the desired number of workers per level while ignoring the short-term capacity reductions induced by training, as in the case of a simple base-stock policy, yields very poor solutions. In contrast, it is sufficient to make decisions with a base/minimum-stock policy when the need for on-the-job training supervision is moderate. If, however, there is a substantial need for supervision significantly decreasing the capacity at the target level, it is best to limit the number of workers who are simultaneously training, as done in the base/minimum-stock/training-capacity policy.

The results further indicate that shorter training periods with lower productivity rates are generally preferable to longer training periods with higher productivity rates. Only when training requires extensive supervision, may it be beneficial to stretch the training duration so as to mitigate the capacity reductions that accompany it. Moreover, the experiments confirmed that for the problem instances examined there is considerable value in downgrading as demonstrated by an average cost decrease of 8.8%. These statistics are consistent with the results reported by Bard (2004); Bard and Purnomo (2005).

With respect to future research, several opportunities exist for expanding the scope of the problem. One example is to allow for dynamic productivity rates over the course of training, where training is more intense in the initial stages and then transitions into an on-the-job mode in the final stages. Furthermore, it is sometimes possible to provide training over a much shortened time period. For example, there are companies that offer to train a foreman in metallurgy in about half a year. If such an opportunity exists, it may be beneficial to release the worker for that time and pay the associated fee. Other extensions might include the incorporation of individual learning rates, learning and forgetting, and individual retirement probabilities. As a word of caution, though, because these extensions require an individual employee view, they are likely pose a tremendous computational burden. From a modeling perspective, including the possibility of hiring experienced workers at high costs would be another interesting aspect. Finally, enhancing the methodology to deal with more complex skill structures is left for further research.

3. A Data-driven Approach for Baggage Handling Operations at Airports

3.1. Introduction

Air passenger volume is continuing to increase, at an average rate of about 5.5% over the last ten years (SITA, 2018). Since existing airport infrastructure can only be expanded in the long-term and at high cost, existing resources need to be used efficiently. A major airline hub requires well-functioning *baggage handling* (BH). If a bag fails to arrive at its destination on time, this reduces customer satisfaction, and the cost of reuniting mishandled bags with their owners is significant, at an average of US$101 and an industry total in the order of US$2.3 billion in 2017 (SITA, 2018).

Central to BH is the *baggage handling system* (BHS), which transports and stores the luggage at airports (see Figure 3.1). Arrival flights deliver inbound luggage that is directed to the baggage claim, and transfer luggage that is directed to departing flights. Typically, inbound luggage is unloaded at stations directly connected to the baggage claim, and the BHS is only involved for transfer luggage or if bags are fed in via a station located remotely. For departing flights, checked-in luggage and transfer luggage is either transferred to the central storage or directly to one of several baggage sorting stations, which are called *carousel*s. A carousel has a circular conveyor belt, from which workers load the baggage into *unit load devices* (ULD) placed on ULD dollies. Once loading is complete, trucks tug the ULD dollies to the airplane, whose departure time determines the end of its BH period.

With several hundred flights per day, BH poses a challenging planning task. While the BHS autonomously transports bags to their destinations, efficient planning is crucial for the *loading process* (LP), i.e. flights have to be assigned to carousels (decision A in Figure 3.1), the loading process and the *storage depletion* (SD) have to be scheduled (decisions B and C), and baggage handlers have to be assigned to load bags into ULDs (decision E). The loading process is very labor intensive, and the high cost pressure for ground handling requires an efficient assignment of the workforce. The goal is to ensure that all bags can be loaded into ULDs in time, that is, before the end of the BH process of the corresponding flight.

In practice, the BH process is typically planned and controlled by a dispatcher using up-to-date information and a graphical user interface. However, without any sophisticated decision support, the combinatorial nature and complexity of the planning task make it impossible to derive good solutions manually. The challenge of the planning task is exacerbated by the need to obtain updated solutions within minutes. Complicating things further, BH is carried out in a dynamic environment with uncertainties about future flight and baggage data. For example, in the year 2017, only 80% of all departures and only 79% of arrivals were on time worldwide (Bureau of Transportation Statistics (BTS), 2018). These unexpected changes, as well as unforeseen arrival times of passengers at the check-in counters, often render carefully conceived plans suboptimal or even infeasible, which can lead to unloaded bags, poor utilization of workers and high costs. Hence, robust planning and dynamic re-optimization are crucial for managing daily operations.

Literature. To date, related literature has primarily focused on managing disruption to the flight schedule, aircraft rotations, crew schedule, passenger itineraries (e.g. Clausen et al., 2010; Petersen et al., 2012; Jiang and Barnhart, 2013), and the gate assignments problem (e.g. Yan and Tang, 2007; Bolat, 2000). Although it is one of an airport's major and most challenging tasks, to date little attention has been paid to the BH at airports. While, to the best of my knowledge, no previous work has been performed on dynamic BH planning, several studies consider a static planning of the baggage handling for deterministic flight schedules. However, due to flight delays the resulting plans are very likely to become infeasible during the day of operations. Abdelghany et al. (2006) propose a greedy based sequential allocation heuristic for assigning prescheduled flights to baggage sorting stations (decision A). Ascó et al. (2014) extend the former work and investigate

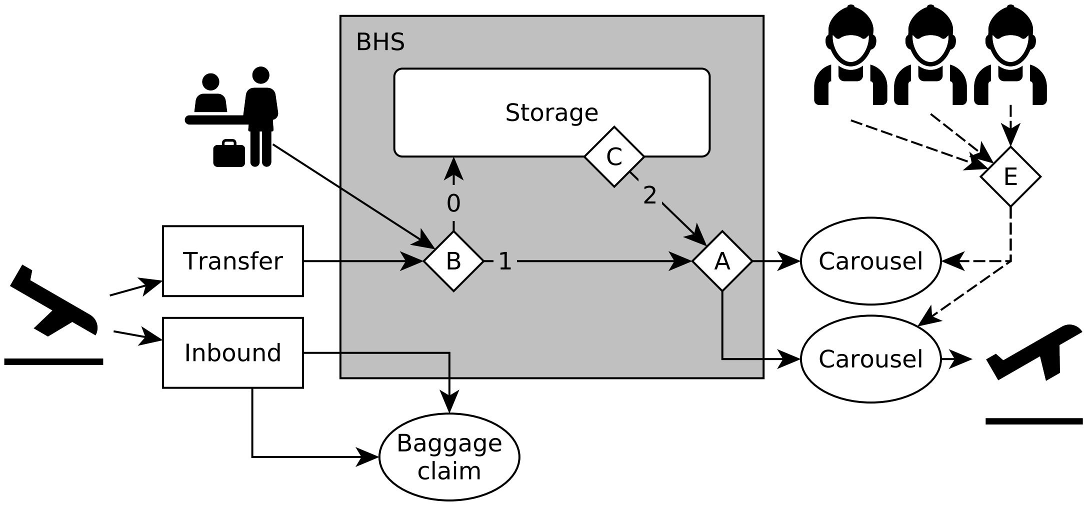


Figure 3.1.: Baggage handling system

several greedy allocation heuristics and a genetic algorithm with respect to multiple objectives, such as minimizing the distance between the assigned baggage sorting station and the aircraft's parking position, called the *stand*, maximizing the buffer times between two consecutive flights on the same baggage sorting station, and fairly distributing the workload across sorting stations. However, neither of them consider the scheduling of the loading process and storage (decisions B and C), which is a key element of a BHS. For example, at Munich Airport, more than 32% of all bags are stored. Nor do they consider efficient assignment of the limited workforce (decision E), which is an important factor for ground handling, in attempts to decrease labor costs (see Barbot, 2012). Frey et al. (2017) present a model that considers both the assignment of flights to carousels and the scheduling of flights' baggage handling (decisions A, B and C) based on point estimates for the baggage arrival processes. While these plans are very unlikely to be feasible on the day of operations, they make it possible to determine the required workforce for the next day, which in turn is an input in the intra-day problem that we consider. A simplifying assumption by Frey et al. (2017) that must be relaxed for the intra-day planning is that the number of workers assigned to a flight must be constant for the complete duration of the loading process. Loading may take up to several hours, and consequently, the number of handlers loading bags may change throughout a flight's loading process. Flexible assignments allow the loading capacity to be matched to fluctuating baggage arrivals more efficiently, and hence a better utilization of the workforce.

Data-driven re-planning. The purpose of this work is to control the baggage handling processes on the day of operation, to avoid unloaded bags. To achieve smooth processing, interrelated decisions need to be updated regularly, taking into account uncertainty and adjusted data. I employ a data-driven approach, i.e. uncertainty is dealt with by directly exploiting data in the optimization. This is in contrast to (i) deterministic optimization based on point estimates, (ii) stochastic programming, where uncertainty is represented by probability distributions, which are either assumed to be known or estimated in a first step, and (iii) robust optimization, where uncertainty is characterized by set membership, rather than distributions, and optimization is against a worst case in this set (Jaillet and Wagner, 2014). Re-optimization is triggered whenever problem parameters change and the computations are carried out while the actual processing of the BH continues. Since decisions have long term consequences and baggage arrivals are uncertain, plans need to be robust. In general, robustness refers to approaches that are immune to data uncertainty (Bertsimas and Sim, 2004), and here robustness means that plans are designed such that failures such as unloaded bags and backlogs in the BHS are unlikely to occur in the presence of uncertain flight data. Robustness is achieved by generating a plan that is feasible for multiple scenarios. The hope is that the actual flight data will not deviate too much from the scenarios for which the generated plan is feasible. To solve each optimization problem quickly, as required by the real-time environment, a column-generation-based heuristic is employed. To obtain integral solutions, column generation is embedded into a branch-and-price search, and a heuristic search strategy called *limited discrepancy search* (see Harvey and Ginsberg, 1995) is used.

The BH problem is formulated as a mathematical model, which allows flexible work profiles and is designed to create robust solutions based on past flight data. The main contributions in this chapter are: (i) a column-generation-based solution method to quickly obtain high-quality solutions, and (ii) management insights regarding the benefits of flexible work profiles based on real-world data. The empirical results, based on a real-world data from Munich Airport, show that the developed procedure outperforms a greedy allocation heuristic that mimics the behavior of a human dispatcher in terms of mishandled bags and capacity violations. Furthermore, it is shown that flexible work profiles significantly improve the system performance. The highest degree of flexibility allows plans to be generated using about half the resources available in reality, and almost entirely without unloaded bags or capacity violations.

The remainder of this chapter is structured as follows: Section 3.2 describes the dynamic BH problem in detail. Section 3.3 presents the dynamic planning approach. The mathematical model is presented in Section 3.4, and the column-generation-based solution procedure is developed in Section 3.5. The computational study in Section 3.6 demonstrates the performance of the proposed procedure in comparison to a greedy approach, and further elaborates the benefits of worker flexibility. Finally, I draw conclusions in Section 3.7.

3.2. Problem Description

Planning the BH involves several interrelated decisions. For each departing flight, we need to (i) assign a carousel, which may be used for several flights simultaneously providing its capacities are not exceeded (decision A in Figure 3.1); (ii) determine the start of the loading process, which is at earliest three hours before a flight's departure (decision B); (iii) set the start of the depletion of the central storage (decision C); and (iv) select a work profile, i.e. determine how many workers should be loading bags at any time during the loading process (decision E). The baggage handling ends about 15 minutes before the estimated departure. While adjusting the start of the loading process and the start of storage depletion influences the arrival of bags at the carousels, adjusting the number of ground handlers influences the speed at which bags are moved from the carousel's conveyor belt into ULDs.

Controlling baggage flows. Before the *loading process* (LP) has started, all incoming bags are sent to the central storage shared by all flights, which has a limited capacity that cannot be exceeded (see Arc 0 in Figure 3.1). At Munich Airport, the storage capacity is 6,000 bags, which is about 10% of the daily luggage. Once the LP has started, all incoming bags are directly sent to the assigned *carousel* (Arc 1). Its capacity is defined as the number of bags that can be placed on its conveyor belt at a time. The *storage depletion* (SD) can begin, at the earliest, when the LP starts. Once the SD has started, the stored bags are transferred to the assigned carousel at a constant rate until all bags for that flight are depleted (Arc 2). Therefore, the arrival of bags at the assigned carousel can be postponed by delaying the start of the LP and the start of the SD. If a carousel's or the storage's capacity is completely used up, bags can no longer be forwarded to the carousel or storage, respectively, and instead arriving bags remain within the BHS until capacity is freed. Such backlogs can hinder the baggage transport for other flights, and such situations must, therefore, be avoided.

Controlling the loading process. Baggage is loaded from the carousels into ULDs by ground handling workers, and since their shift schedules are fixed ahead of the planning day, limited personnel is available. A carousel has up to six working stations supporting baggage handlers to load bags into the correct ULDs. Each working station consists of a screen showing which bag belongs in which ULD, and a hand-held scanner to capture each bag's tag and its ULD. For security reasons and to reduce the number of mishandled bags, one worker exclusively uses one working station and handles one flight at a time. A constant loading rate for each worker per period is assumed. The number of workers assigned to a flight can change during the loading process, and planners often decide to allocate a certain number of workers for the entire time the flight is processed and additional workers during the peak periods. However, it is impractical to reassign workers too often, because coordinating the workers would get too difficult. Therefore, I introduce the *minimum block length*, which defines a timespan for which the number of assigned workers must be constant. The minimum block length concept has been introduced for the resource-constrained project scheduling problem with flexible work profiles in Fündeling and Trautmann (2010). Additionally, I introduce the *maximum number of changes*, which defines the maximum number of times the number of workers is allowed to change during a flight's loading process. This allows a constant number of workers to be enforced for the entire duration of the loading process. Minimum block length and maximum number of changes will allow to analyze the value of worker flexibility in detail.

Flight departures. BH ends 15 minutes before the flight's estimated departure time, but departures are often delayed. If delays are substantial and anticipated well in advance, the end of the BH is updated accordingly, which can be taken into account in the dynamic planning procedure developed in this work. Delays that occur on short notice due to unexpected problems, such as when refueling or cleaning takes too long, mechanical breakdowns, or missing crew, do not delay the end of the BH. For simplicity, it is assumed that the end of the BH is not re-scheduled within the last 30 minutes before the flight's departure.

Flight arrivals. The *on-block time* (arrival time of airplane at its stand) of each inbound flight may be ahead of plan or delayed, and the deviation depends on factors such as airspace and airport capacity. Within this, the airport's runway capacity, which depends on weather conditions, is the most crucial factor (see Ball et al., 2007). Another aspect influencing the on-block time is the *aircraft landing sequence*. While most airports sequence landings according to the first-come, first-served discipline, some optimize the sequence (e.g. Beasley et al., 2000; Balakrishnan and Chandran, 2010), which may cause a deviation from the estimated on-block time on short notice. In general, the on-block time can be predicted quite accurately, and for simplicity, it is assumed that the on-block time becomes known 30 minutes ahead of time.

Check-in baggage arrival processes. The number of bags arriving over time is uncertain — for both check-in baggage and transfer baggage. The arrival process of check-in luggage is uncertain, since passengers' arrival times at the check-in counters, the amount of luggage they carry, and the processing times at check-in are all uncertain. However, arrivals of passengers at the check-in counters show similar patterns for flights with identical flight numbers and destinations that depart on the same day of the week (see Stolletz, 2011), providing a set of observed check-in arrival processes for each departing flight. Hence, for each departing flight, flights with the same flight number and destination, departing on a different date with the same weekday are grouped to obtain multiple observed check-in arrival processes, which will be exploited in the optimization.

Transfer baggage arrival process. The arrival process of transfer luggage is uncertain, since for each inbound flight, the number of bags, on-block time and duration until bags are fed into the BHS are uncertain. However, the quantity of baggage an inbound flight carries becomes known once its BH is completed at the origin airport, which makes the number of bags delivered to each outbound flight deterministic. For simplicity, it is assumed that the quantity of baggage becomes known 15 minutes before the departure at the origin airport. As soon as the inbound aircraft has reached its stand, its ULDs are unloaded and delivered to loading stations. The amount of time between the on-block time and the point at which bags begin being fed into the BHS varies slightly depending on the locations of the loading station used, and takes an average of 18 minutes, for example, at Munich Airport. It is assumed that the point at which bags begin to be fed into the BHS becomes known at the on-block time. At Munich Airport, it usually takes less than 45 minutes for all bags to be fed into the BHS, and an average of about five minutes for them to be transported through the BHS. The rate at which the bags are fed into the BHS is assumed to be constant.

3.3. Rolling Horizon Approach

The overall aim of the airport is to plan the baggage loading process such that, over the course of a day, the total number of unloaded bags, as well as capacity violations of the carousels' conveyor belts and the storage remain as low as possible. In order to reach this goal, I introduce a rolling horizon approach that (i) exploits the information that becomes known as time progresses, (ii) delivers an optimized plan in each re-optimization, even when resources are insufficient, and (iii) increases the likelihood that plans are feasible in the event of unexpected baggage flows, by exploiting arrival processes that have been observed in the past.

Rolling planning. To profit from updated flight data, the planning is revised in the course of the planning day in a rolling horizon fashion, leading to a sequence of optimization problems. Since solving each optimization problem requires some time, a solution is created for a future point in time. The state of the system at that time results from the current state and the current plan. The state determines whether the LP of a flight is in progress, and if so, at which carousel and whether the SD is in progress. In the meantime, since the baggage flow from now until the time of the future state is uncertain, we do not know exactly how many bags will be in the storage and on the conveyor belt. To increase the tractability of each re-optimization problem, the planning horizon is limited. The planning horizon needs to be chosen sufficiently large because decisions that are too shortsighted may cause backlogs in the BHS and unloaded bags.

Certain and uncertain bags. As decisions have long term consequences and baggage flows are uncertain, re-optimization alone is insufficient. Instead, robust plans need to be generated in each re-optimization. Simply discretizing time and planning based on point estimates for the number of bags arriving per period leads to bad outcomes, since (i) for given point estimates, say the 70%–quantiles, it is often not possible to generate a feasible plan as unloaded bags and capacity violations may be inevitable when resources are scarce, and (ii) minor deviations from the point estimates make the resource allocation inappropriate. Stochastic optimization based on scenarios (e.g. maximizing the number of observed arrival processes for which a solution is feasible) or robust optimization (e.g. optimizing a worst case) is appealing, but makes the problem computationally intractable. Instead, resource-feasible plans that are designed for as much baggage as possible are generated. More precisely, baggage is classified into *certain bags* and *uncertain baggage*. Certain bags are either already in the system or will arrive with certainty. However, the exact arrival time may still be uncertain, i.e. transfer baggage is classified as certain once baggage handling at the origin airport is finished, as it will arrive with certainty, even though the exact time is unknown. Once a baggage tow truck is headed towards a loading station to feed bags into the BHS, it is also known with certainty when those transfer bags will be fed into the BHS. Uncertain baggage comprises baggage streams where the exact amount is uncertain. Using lexicographic ordering, the presented planning approach will strictly prioritize certain baggage over uncertain baggage, which leads to the optimization being carried out in two phases.

Ensuring feasibility. In the first phase, the optimization seeks a feasible plan maximizing the fraction of the certain bags that can be considered for each flight. In order to not discriminate individual flights, we use lexicographic preferences: maximize the smallest fraction of all flights, then maximize the second smallest fraction of all flights, and so on. At the end of the first phase, the certain bags that cannot be handled are disregarded.

Robustness against uncertainty. In the second phase, the uncertain baggage is also included. Past observations are used to generate a set of ordered scenarios for each flight, where for every period the number of incoming bags in a scenario is greater than or equal to the corresponding number in the previous scenario. Note that the non-discarded certain bags from phase 1 are included in every scenario. The optimization seeks a plan that is feasible while maximizing the scenario indices of the flights, thereby increasing the likelihood that unexpected baggage arrival streams will be absorbed. Intuitively, if for example it turns out that a flight can be planned based on the scenario with the largest values, it is likely that everything will work out well. However, it is unlikely that the actual number of incoming bags will reach the maximum values in every period. Instead, the actual values are above average in some periods, while falling below average in other periods, which means that they tend to cancel each other out. Flights compete for resources, and since resources are scarce, it cannot be guaranteed that all bags can be loaded into ULDs for all flights. Therefore, I again use lexicographic preferences: maximize the smallest scenario index used for all flights, then maximize the second smallest index of all flights, and so on.

3.4. Mathematical Model

In this section, the mathematical model for the outbound baggage handling problem is presented. All notation is summarized in Appendix B. Planning is based on a discrete planning horizon $\mathcal{T} = \{0, \ldots, T\}$ that consists of $T+1$ evenly spaced points in time or T periods of equal length. Period t refers to the time interval $[t, t+1)$. The period length is set to 5 minutes in the experiments. Time 0 marks the actual start of the day of operation. However, an algorithm may require some time to generate a solution, and the first solution needs to be available at time 0. Therefore, an optimization problem is solved starting at each $t = -\Delta, \ldots, T-1-\Delta$ with respect to the current state, planning, and updated flight data, where Δ denotes the time limit for optimization in numbers of time periods. Let P^t denote the problem that is solved starting at time t. P^t has to return a solution until $t'_t := t + \Delta$, when the new planning takes effect. For example, the first problem $P^{-\Delta}$ creates a plan starting from time 0 with the information that is available at time $-\Delta$. To increase the tractability of the problem, the length of the planning horizon is limited by parameter Δ^{h}. The planning horizon of P^t is denoted as $\mathcal{T}_t := \left\{t'_t, \ldots, t'_t + \Delta^{\mathrm{h}}\right\}$ with $\Delta^{\mathrm{h}} \geq 1$.

Flights. The set of departing flights of the complete planning day is denoted as $\mathcal{F}$. The end of flight i's baggage handling is denoted as E_i. Each optimization problem P^t has to consider the state of the system at time t'_t, which depends on the state of the system now and the plan from now until t'_t. Therefore, flights are categorized into the following subsets, of which only those needed in the model are formally defined. The flights that are finished by t'_t as well as the flights that can start at earliest at $t'_t + \Delta^{\mathrm{h}}$ do not need to be considered and are excluded from the decision making. Let $\mathcal{F}^t$ denote the subset of flights of $\mathcal{F}$ that are included in P^t. They are categorized into *flights in progress*, whose loading process has or will be started by time t'_t (SD may or may not have started), *planned flights*, for which a plan is available from the previous optimization, but the loading process has not been started, and *new flights*, which are considered for the first time in P^t (i.e. they were not considered in the previous optimization).

Carousels. Let $\mathcal{C}$ denote the set of carousels. A carousel $c \in \mathcal{C}$ is characterized by the conveyor belt capacity measured in number of bags K_c^{b} and the number of working stations K_c^{w}. Once a flight's loading process has started at a carousel, it has to be finished at the same carousel. Let $\mathcal{C}_i$ denote the set of carousels flight i can be assigned to.

Baggage arrival streams. For each flight, the model will use scenarios of baggage arrival streams, where each scenario comprises two sub-streams, one for *certain bags* and and one for *uncertain baggage*. Accordingly, a scenario is referenced by a pair of indices (k, k'), where k refers to the certain bags and is from index set $\mathcal{K}^{\mathrm{c}} := \{1, \ldots, K^{\mathrm{c}}\}$ and k' refers to the uncertain baggage and is from index set $\mathcal{K}^{\mathrm{u}} := \{0, \ldots, K^{\mathrm{u}}\}$. Let $A_{i\tau}^t$ denote the number of certain bags arriving in period $\tau = -1, \ldots, E_i - 1$ for flight $i \in \mathcal{F}$ at the carousel or the storage as determined at time t. Period $\tau = -1$ is reserved for bags that have arrived ahead of the planning day and that are already in the central baggage storage at the beginning of the planning horizon. The number of certain bags arriving at their destination in scenario (k, k') (for t, i, τ as previously) is calculated as $\lfloor \alpha_k A_{i\tau}^t + \frac{1}{2} \rfloor$, where $0 \leq \alpha_1 < \cdots < \alpha_{K^{\mathrm{c}}} = 1$. The number of uncertain bags arriving at its destination in scenario (k, k') (for t, i, τ as previously) is denoted as $A_{i\tau}^{tk'}$. It consists of the $q_{k'}$-quantile of the empirical bag distribution of that period, where $q_1 < q_2 < \ldots < q_{K^{\mathrm{u}}} = 100\%$. For $k' = 0$ we set $A_{i\tau}^{tk'} = 0$. The empirical bag distributions are obtained from a number of observations of the same flight on the same weekday, e.g. eight observations of the United flight from Munich to New York on Tuesday at 12:35 pm. Now the certain bags are combined with the uncertain baggage: let $A_{i\tau}^{tkk'} = \lfloor \alpha_k A_{i\tau}^t + \frac{1}{2} \rfloor + A_{i\tau}^{tk'}$ denote the number of bags arriving at its destination in *scenario* $(k, k') \in \mathcal{K}^c \times \mathcal{K}^u$ (for t, i, τ as previously). *Scenario* (k, k') of flight i at time t is defined as the vector $\left(A_{i\tau}^{tkk'}\right)_{\tau=-1,\dots,E_i-1}$. By the construction of the scenarios $A_{i\tau}^{t,1,k'} \leq A_{i\tau}^{t,2,k'} \leq \dots \leq A_{i\tau}^{tK^{\mathrm{c}}k'}$ and $A_{i\tau}^{t,k,1} \leq A_{i\tau}^{t,k,2} \leq \dots \leq A_{i\tau}^{tkK^{\mathrm{u}}}$ hold. Let me give the following illustrative example where indices t, i and τ are dropped. Let the number of certain bags be $A = 5$ and the empirical distribution of the uncertain bags defined by all observations be $(0, 1, 2, 2, 2, 3, 3, 5)$. With $\alpha_2 = 0.5$ and $q_3 = 0.75$ we obtain $A^{2,3} = \left\lfloor 0.5 \cdot 5 + \frac{1}{2} \right\rfloor + 3 = 6$ bags for scenario $(2, 3)$.

Schedules and work profiles. The loading process of flight i can start, at the earliest, at $S_i \in \mathcal{T}$, which is set to $S_i = E_i - 36$ in the experiments, i.e. the LP can start 3 hours before the end of the BH. The start of the LP and of the storage depletion of flight $i \in \mathcal{F}$ are denoted as $s_i^{\mathrm{slp}} \in \mathcal{T}$ and $s_i^{\mathrm{ssd}} \in \mathcal{T}$ (with $s_i^{\mathrm{ssd}} \geq s_i^{\mathrm{slp}}$), respectively. Additionally, the following indicators will be useful: $s_{i\tau}^{\mathrm{lp}} \in \{0, 1\}$ and $s_{i\tau}^{\mathrm{sd}} \in \{0, 1\}$ for $\tau = S_i, \dots, E_i - 1$ are equal to one if the LP and the SD of flight i are in progress in period τ and zero otherwise, respectively. Let $w_{i\tau}$ denote the number of assigned workers for flight i in period $\tau = S_i, \dots, E_i - 1$, which is equal to the number of used working stations of the corresponding carousel. Workers must not be assigned before the loading process has started, and due to organizational regulations, at least one worker must be assigned throughout the entire loading process.

The block length L defines a number of consecutive periods that must have a constant number of workers, i.e. if $w_{i,\tau-1} \neq w_{i\tau}$, then $w_{i\tau} = \ldots = w_{i,\tau+L-1}$, where it is assumed that $w_{i,S_i-1} = 0$ and $w_{iE_i} = 0$. The maximum number of changes J limits how often the number of workers assigned to a flight is allowed to change during its loading process. A schedule and profile for flight i is denoted as

$$\pi = \left(s_\pi^{\mathrm{slp}}, \left(s_{\tau\pi}^{\mathrm{lp}}\right)_{\tau=S_i,\dots,E_i-1}, \left(w_{\tau\pi}\right)_{\tau=S_i,\dots,E_i-1}, s_\pi^{\mathrm{ssd}}, \left(s_{\tau\pi}^{\mathrm{sd}}\right)_{\tau=S_i,\dots,E_i-1}\right)$$

where index i at π indicating the flight is omitted, and it is referred to as schedule or profile, depending on the context.

Storage. The central storage has a capacity of K^{s} bags. Stored bags are transferred to the assigned carousel at a constant rate of r^{s} bags per time period. The number of bags in storage for flight i and its scenario (k, k') at time $\tau = 0, \ldots, E_i$, given profile π, can be calculated recursively as follows.

$$\phi^{\mathrm{s},tkk'}_{i\tau\pi} = \begin{cases} A^{tkk'}_{i,-1} & \text{for } \tau = 0 & (a) \\ \phi^{\mathrm{s},tkk'}_{i,\tau-1,\pi} + A^{tkk'}_{i,\tau-1} & \text{for } 0 < \tau \leq s^{\mathrm{slp}}_{\pi} & (b) \\ \phi^{\mathrm{s},tkk'}_{i,\tau-1,\pi} & \text{for } s^{\mathrm{slp}}_{\pi} < \tau \leq s^{\mathrm{ssd}}_{\pi} & (c) \\ \left[\phi^{\mathrm{s},tkk'}_{i,\tau-1,\pi} - r^{\mathrm{s}}\right]^{+} & \text{for } s^{\mathrm{ssd}}_{\pi} < \tau, & (d) \end{cases} \tag{3.1}$$

where $[x]^{+}$ is used as a short form for $\max\{x, 0\}$. Equation (3.1) captures the baggage flows affecting the storage. Bags arriving before the loading process starts are directed to the storage (a and b). The bags remain in the storage until the depletion starts (c) and are depleted at a constant rate once the depletion has started (d).

Carousel load. Let $\phi^{\mathrm{s}\to\mathrm{b},tkk'}_{i\tau\pi} = \left[\phi^{\mathrm{s},tkk'}_{i\tau\pi} - r^{\mathrm{s}}\right]^{+} = \phi^{\mathrm{s},tkk'}_{i,\tau+1,\pi} - \phi^{\mathrm{s},tkk'}_{i\tau\pi}$ denote the number of bags of flight i that are transferred from the central storage to the carousel in period $\tau = S_i, \ldots, E_i - 1$ given schedule π and scenario (k, k'). The number of available workers during period τ is denoted as K^{wo}_{τ}, and the loading rate per worker (and working station) is r^{l} bags per period, which is assumed to be constant. The number of bags on the carousel's conveyor belt for flight i and its scenario (k, k') at time $\tau = S_i, \ldots, E_i$, given profile π, can be calculated recursively as follows.

$$\phi^{\mathrm{b},tkk'}_{i\tau\pi} = \begin{cases} 0 & \text{for } \tau \leq s^{\mathrm{slp}}_{\pi} & (a) \\ \left[\phi^{\mathrm{b},tkk'}_{i,\tau-1,\pi} + A^{tkk'}_{i,\tau-1} - w_{\pi,\tau-1} r^{\mathrm{l}}\right]^{+} & \text{for } s^{\mathrm{slp}}_{\pi} < \tau \leq s^{\mathrm{ssd}}_{\pi} & (b) \\ \left[\phi^{\mathrm{b},tkk'}_{i,\tau-1,\pi} + A^{tkk'}_{i,\tau-1} - w_{\pi,\tau-1} r^{\mathrm{l}} + \phi^{\mathrm{s}\to\mathrm{b},tkk'}_{i,\tau-1,\pi}\right]^{+} & \text{for } s^{\mathrm{ssd}}_{\pi} < \tau. & (c) \end{cases} \tag{3.2}$$

Equation (3.2) captures the baggage flows affecting the number of bags on the assigned carousel's conveyor belt. Before the loading process has started, there are no bags on the carousel as they are directed to the storage instead (a). From the start of the loading process, arriving bags are directed to the carousel, and the assigned workers load bags from the carousel into containers (b). Once storage depletion has begun, the bags that arrive from the storage are added as well (c).

Feasible profiles. At time t, a profile π for flight i is feasible at carousel $c \in \mathcal{C}_i$ with respect to scenario (k, k') if the carousel capacity is not exceeded, i.e. if $\phi^{\mathrm{b},tkk'}_{i\tau\pi} \leq K^{\mathrm{b}}_c$ for all $\tau = S_i, \ldots, E_i - 1$, and if there are no unloaded bags at time E_i, i.e. if both $\phi^{\mathrm{s},tkk'}_{iE_i\pi}$ and $\phi^{\mathrm{b},tkk'}_{iE_i\pi}$ are zero. Let $\mathcal{S}^{tkk'}_{ic}$ denote the set of feasible profiles for $i \in \mathcal{F}$, $c \in \mathcal{C}_i$, $k \in \mathcal{K}^{\mathrm{c}}$, $k' \in \mathcal{K}^{\mathrm{u}}$ and $t \in \mathcal{T}$. Furthermore, set $\mathcal{S}^{tkk'}_{ic\tau} := \left\{\pi \in \mathcal{S}^{tkk'}_{ic} \mid s^{\mathrm{slp}}_{\pi} \leq \tau < E_i\right\}$ is the subset of $\mathcal{S}^{tkk'}_{ic}$ where flight i's loading process is in progress during period τ.

Model formulation. Each optimization problem P^t finds a resource-feasible solution for the planning horizon $\mathcal{T}_t$ with the objective to consider as large as possible a number of certain bags and the highest possible quantiles of uncertain baggage, thereby increasing the robustness. The index t indicating the current time is dropped in the model. Binary variable $x^{kk'}_{ic\pi}$ is one if the loading process of flight $i \in \mathcal{F}^t$ is conducted at carousel $c \in \mathcal{C}_i$ with profile $\pi \in \mathcal{S}^{tkk'}_{ic}$, and zero otherwise. Planning flight i with a profile $\pi \in \mathcal{S}^{kk'}_{ic}$ and hence with scenario (k, k') incurs costs of $p^{\mathrm{c}}_k + p^{\mathrm{u}}_{k'}$, which are set such that $p^{\mathrm{u}}_{K^{\mathrm{u}}} = 1$, $|\mathcal{F}^t|\, p^{\mathrm{u}}_k \leq p^{\mathrm{u}}_{k-1}$ for $k = K^{\mathrm{u}}, \ldots, 1$, $|\mathcal{F}^t|\, p^{\mathrm{u}}_0 \leq p^{\mathrm{c}}_{K^{\mathrm{c}}}$ and $|\mathcal{F}^t|\, p^{\mathrm{c}}_k \leq p^{\mathrm{c}}_{k-1}$ for $k = K^{\mathrm{c}}, \ldots, 2$ holds. The scenarios are, therefore, prioritized in the order $(1, k'), \ldots, (K^{\mathrm{c}}, k'')$ independent of k' and k'', and $(k, 0), \ldots, (k, K^{\mathrm{u}})$ for a fix k. Hence, for example, improving from scenario $(1, 0)$ to $(2, 0)$ for a single flight decreases the objective value more than improving from scenario $(2, 0)$ to $(3, 0)$ for all other flights. Problem P^t can be stated as follows:

$$
\begin{aligned}
&(P^t) \quad \min \quad \sum_{i\in\mathcal{F}}\sum_{c\in\mathcal{C}_i}\sum_{k\in\mathcal{K}^{\mathrm{c}}}\sum_{k'\in\mathcal{K}^{\mathrm{u}}}\sum_{\pi\in\mathcal{S}_{ic}^{kk'}} (p_k^{\mathrm{c}}+p_k^{\mathrm{u}})\, x_{ic\pi}^{kk'} && (3.3)\\
&\text{subject to}\\
&\sum_{c\in\mathcal{C}_i}\sum_{k\in\mathcal{K}^{\mathrm{c}}}\sum_{k'\in\mathcal{K}^{\mathrm{u}}}\sum_{\pi\in\mathcal{S}_{ic}^{tkk'}} x_{ic\pi}^{kk'} = 1 && \forall i\in\mathcal{F} && (3.4)\\
&\sum_{i\in\mathcal{F}}\sum_{k\in\mathcal{K}^{\mathrm{c}}}\sum_{k'\in\mathcal{K}^{\mathrm{u}}}\sum_{\pi\in\mathcal{S}_{ic\tau}^{kk'}} w_{\pi\tau} x_{ic\pi}^{kk'} \le K_c^{\mathrm{w}} && \forall c\in\mathcal{C},\tau\in\mathcal{T} && (3.5)\\
&\sum_{i\in\mathcal{F}}\sum_{k\in\mathcal{K}^{\mathrm{c}}}\sum_{k'\in\mathcal{K}^{\mathrm{u}}}\sum_{\pi\in\mathcal{S}_{ic\tau}^{kk'}} \phi_{i\tau\pi}^{\mathrm{b},tk} x_{ic\pi}^{kk'} \le K_c^{\mathrm{b}} && \forall c\in\mathcal{C},\tau\in\mathcal{T} && (3.6)\\
&\sum_{i\in\mathcal{F}}\sum_{c\in\mathcal{C}_i}\sum_{k\in\mathcal{K}^{\mathrm{c}}}\sum_{k'\in\mathcal{K}^{\mathrm{u}}}\sum_{\pi\in\mathcal{S}_{ic}^{kk'}} \phi_{i\tau\pi}^{\mathrm{s},tk} x_{ic\pi}^{kk'} \le K^{\mathrm{s}} && \forall \tau\in\mathcal{T} && (3.7)\\
&\sum_{i\in\mathcal{F}}\sum_{c\in\mathcal{C}_i}\sum_{k\in\mathcal{K}^{\mathrm{c}}}\sum_{k'\in\mathcal{K}^{\mathrm{u}}}\sum_{\pi\in\mathcal{S}_{ic\tau}^{kk'}} w_{\pi\tau} x_{ic\pi}^{kk'} \le K_\tau^{\mathrm{wo}} && \forall \tau\in\mathcal{T} && (3.8)\\
&x_{ic\pi}^{kk'} \in \{0,1\} && \forall i\in\mathcal{F}, c\in\mathcal{C}_i && (3.9)\\
& && k\in\mathcal{K}^{\mathrm{c}}, k'\in\mathcal{K}^{\mathrm{u}},\\
& && \pi\in\mathcal{S}_{ic}^{kk'}
\end{aligned}
$$

Objective function (3.3) penalizes the use of scenarios with low indices. Partitioning constraints (3.4) require the assignment and scheduling of each flight that is included in the current planning period. Constraints (3.5) limit the number of available working stations. Carousels' conveyor belt capacities are limited in constraints (3.6). The central storage's capacity is limited by constraints (3.7). Constraints (3.8) make sure that worker capacities are not exceeded. Finally, constraints (3.9) define the variables to be binary.

3.5. Solution Methodology

Model (3.3) — (3.9) is $\mathcal{NP}$–complete ($\mathcal{NP}$–hard by reduction from the set partitioning problem and in $\mathcal{NP}$) and suffers from the defect that the variables, of which there are an exponential number, should be enumerated. For each flight $i \in \mathcal{F}^t$, carousel $c \in$

$\mathcal{C}_i$, and scenario $(k,k') \in \mathcal{K}^{\mathrm{c}} \times \mathcal{K}^{\mathrm{u}}$ the number of profiles $\left|\mathcal{S}_{ic}^{tkk'}\right|$ is in $\mathcal{O}\left(\left(K_c^{\mathrm{w}}\right)^{E_i-S_i}\right)$. Since most of these variables will be zero in an optimal solution, promising variables, i.e. new profiles, are generated as needed by means of column generation. The master problem corresponds to problem P^t and there are $\mathcal{O}\left(|\mathcal{F}^t| \cdot |\mathcal{C}| \cdot |\mathcal{K}^{\mathrm{c}}| \cdot |\mathcal{K}^{\mathrm{u}}|\right)$ subproblems — one per flight, carousel, and scenario. The subproblems will be solved efficiently with dynamic programming as shown below. To obtain integral solutions, column generation is embedded into branch-and-price. Since reaching and proving optimality can take too long, a heuristic search strategy is applied. The lexicographic objective (3.3) can be minimized in stages, which avoids numerical problems that otherwise occur due to the large range of objective coefficients $[p^{\mathrm{u}}_{K^{\mathrm{u}}}, p^{\mathrm{c}}_1]$ and reduces the number of variables that must be considered simultaneously, which increases the tractability of the problem significantly.

3.5.1. Minimizing the lexicographic objective

The lexicographic objective (3.3) can be minimized in two phases consisting of K^{c} and K^{u} stages, respectively. In the first phase, a solution is found for scenarios $(1,0), \ldots, (K^{\mathrm{c}},0)$, that is, ignoring the uncertain baggage. In the first stage of the first phase, a feasible solution must be found for scenario $(1,0)$ of all flights. At each subsequent stage $k = 2, \ldots, K^{\mathrm{c}}$, there is a set of "open flights" and a set of "fixed flights", in which each fixed flight i is fixed to a scenario $\left(\hat{k}_i, 0\right)$ with $\hat{k}_i < k$. Initially, all flights are open and the set of fixed flights is empty. In stage k, the optimization attempts to use scenario $(k,0)$ for the open flights subject to using scenario $\left(\hat{k}_i, 0\right)$ for all fixed flights, and when the optimization terminates with an integral solution, each open flight i that is not planned with scenario $(k,0)$ is moved from the set of open flights to the set of fixed flights with $\hat{k}_i = k-1$. Afterwards, the procedure moves on to the next stage $k \leftarrow k+1$. Proceeding this way, only the variables $x_{ic\pi}^{\hat{k}_i 0}$ are needed for the fixed flights, and penalty costs are unnecessary, and only the variables $x_{ic\pi}^{k-1,0}$ and $x_{ic\pi}^{k,0}$ are needed for the open flights, where all $x_{ic\pi}^{k-1,0}$ are penalized. Note that the solution of the previous stage is always feasible for the current stage. At the end of the first phase, $\hat{k}_i$ is fixed to K^{c} for all flights that remain open. In the second phase, the same procedure is applied for the uncertain baggage, that is, the procedure iteratively tries to increase the scenarios of the open flights from $\left(\hat{k}_i, k'-1\right)$ to $\left(\hat{k}_i, k'\right)$ for $k' = 1, \ldots, K^{\mathrm{u}}$ subject to using scenario $\left(\hat{k}_i, \hat{k}'_i\right)$ for all fixed flights.

3.5.2. Column generation

Problem P^t is the *master problem* (MP) of the column generation. Since there is an exponential number of profiles, the sets of profiles are restricted to subsets $\tilde{\mathcal{S}}_{ic}^{tkk'} \subseteq \mathcal{S}_{ic}^{tkk'}$. The MP containing only the columns in $\bigcup_{i \in \mathcal{F}^t, c \in C, k \in \mathcal{K}^c, k' \in \mathcal{K}^u} \tilde{\mathcal{S}}_{ic}^{tkk'}$ is called the *restricted master problem* (RMP). As the RMP does not necessarily yield an optimal solution to the MP, the integrality requirements on the variables are relaxed and the continuous relaxation of RMP is solved to optimality. Then, the optimal dual variable values are used to identify new variables with negative reduced costs, which is done in subproblems called *oracles* or *pricing problems* — one for each flight-carousel-scenario triplet. If a column with negative reduced costs is found, it is added to the RMP and the process repeats until no column with negative reduced costs can be found in any subproblem. As then there is no candidate column that would improve the objective function value, the solution is optimal for the linear relaxation of MP.

Pricing refers to any method of computing the reduced costs vector of the non-basic variables. However, it is not necessary to compute the reduced costs of all non-basic variables. Pricing out only a few candidates for entering the basis is known as *partial pricing* (Chvátal, 1983). Column generation can be considered a (partial) *pricing scheme* for large-scale linear programs (see Lübbecke and Desrosiers, 2005). When there are many subproblems, as is the case here, it is often more efficient to consider only a few of them in each iteration, which is called *partial column generation*. When choosing a pricing scheme, it is important to balance the computational time spend for the subproblems and the RMP. On the one extreme, when all subproblems are solved and only the best column is added to the RMP, a lot of time is spent on solving the subproblems, but the RMP receives relatively few high-quality columns. On the other hand, if more columns are added to the RMP, it becomes harder to solve, but fewer iterations may be needed. Here, it is best to solve all pricing problems and add the best column, as the pricing problems can be solved efficiently. As every subproblem runs in a single threat, the pricing problems can be solved concurrently, which speeds up each iteration further.

Decision variables for $PP_{ic}^{tkk'}$	
$s_\tau^{\mathrm{lp}} \in \{0,1\}$	1, if the loading process is in progress in period $\tau = \bar{S}_i - 1, \ldots, E_i - 1$ and 0 otherwise
$s_\tau^{\mathrm{sd}} \in \{0,1\}$	1, if the storage depletion is in progress in period $\tau = \bar{S}_i - 1, \ldots, E_i - 1$ and 0 otherwise
$w_\tau \in \{0, \ldots, K_c^{\mathrm{w}}\}$	Number of workers (and working stations) in period $\tau = \bar{S}_i - 1, \ldots, E_i - 1$
$\delta_\tau \in \{1, \ldots, L\}$	Number of periods with constant number of workers at time $\tau = \bar{S}_i, \ldots, E_i$ before a decision is made, i.e. excluding a potential change at time τ
$\gamma_\tau \in \{1, \ldots, J\}$	Number of times the number of workers has changed up to time $\tau = \bar{S}_i, \ldots, E_i$ before a decision is made
$\phi_\tau^{\mathrm{s}} \in \left\{0, \ldots, \sum_\tau A_{i\tau}^{tkk'}\right\}$	Number of bags in the central storage at time $\tau = \bar{S}_i, \ldots, E_i$
$\phi_\tau^{\mathrm{b}} \in \left\{0, \ldots, K_c^{\mathrm{b}}\right\}$	Number of bags on the conveyor belt at time $\tau = \bar{S}_i, \ldots, E_i$

Table 3.1.: Decision variables for pricing problem $PP_{ic}^{tkk'}$

3.5.3. Subproblems

Assume the column generation procedure is at any iteration and the continuous relaxation of RMP is solved. Let λ_i^{a}, $\lambda_{c\tau}^{\mathrm{w}}$, $\lambda_{c\tau}^{\mathrm{b}}$, $\lambda_\tau^{\mathrm{s}}$, and $\lambda_\tau^{\mathrm{wo}}$ denote the dual variable values of the assignment constraints (3.4), and the capacity constraints (3.5), (3.6), (3.7), and (3.8), respectively. The most promising schedule for flight $i \in \mathcal{F}^t$, carousel $c \in \mathcal{C}_i$, and scenario $(k, k') \in \mathcal{K}^{\mathrm{c}} \times \mathcal{K}^{\mathrm{u}}$ is the solution of the *pricing problem* $PP_{ic}^{tkk'}$, which is a shortest path problem from a source node to a destination node (one-to-one) on an acyclic network. Hence, $PP_{ic}^{tkk'}$ can be solved efficiently with dynamic programming. Let $\bar{S}_i := \max\{S_i, t'_t\}$ denote the first period of $PP_{ic}^{tkk'}$. The decision variables are defined in Table 3.1. The indices i, c, k, k' and π are omitted.

System states. The state at time $\tau = \bar{S}_i, \ldots, E_i$ is defined as

$$z_\tau = \left(s_{\tau-1}^{\mathrm{lp}}, w_{\tau-1}, \delta_\tau, \gamma_\tau, s_{\tau-1}^{\mathrm{sd}} \phi_\tau^{\mathrm{s}}, \phi_\tau^{\mathrm{b}}\right)$$

and refers to time τ immediately before a decision is made. The initial state $z_{\bar{S}_i}$ at time $\bar{S}_i$ depends on the current state of the flight, that is, if the flight's LP and SD are in progress and how many bags are in the storage and on the carousel's belt at time $\bar{S}_i$, respectively.

Transitions. When making a decision $a_\tau \coloneqq \left(s^{\mathrm{lp}}_\tau, s^{\mathrm{sd}}_\tau, w_\tau\right)$ at time $\tau = \bar{S}_i, \ldots, E_i - 1$ given state $z_\tau = \left(s^{\mathrm{lp}}_{\tau-1}, w_{\tau-1}, \delta_\tau, \gamma_\tau, s^{\mathrm{sd}}_{\tau-1}\phi^{\mathrm{s}}_\tau, \phi^{\mathrm{b}}_\tau\right)$, the following constraints must hold.

$$s^{\mathrm{lp}}_{\tau-1} \leq s^{\mathrm{lp}}_\tau \tag{3.10}$$

$$s^{\mathrm{sd}}_{\tau-1} \leq s^{\mathrm{sd}}_\tau \tag{3.11}$$

$$s^{\mathrm{sd}}_\tau \leq s^{\mathrm{lp}}_\tau \tag{3.12}$$

$$s^{\mathrm{lp}}_\tau \leq w_\tau \leq K^{\mathrm{w}}_c s^{\mathrm{lp}}_\tau \tag{3.13}$$

$$\delta_\tau < L \Rightarrow w_\tau = w_{\tau-1} \tag{3.14}$$

$$w_{E_i-L} = \ldots = w_{E_i-1} \tag{3.15}$$

$$\gamma_\tau = J \Rightarrow w_\tau = w_{\tau-1} \tag{3.16}$$

Constraints (3.10) and (3.11) state that both the loading process and the storage depletion cannot be interrupted once started. Constraint (3.12) makes sure that the storage depletion does not start before the loading process. Constraint (3.13) requires at least one worker to be assigned during the loading process, and at the most as many as the carousel has working stations, and workers may not be assigned before the loading process has started. Constraints (3.14) and (3.15) enforce the block length restriction, where the latter affects the end of the loading process. Finally, constraint (3.16) limits the number of times the number of workers can change.

The next state $z_{\tau+1}$ results from the current state z_τ and the decision a_τ. The state variables s^{lp}_τ, s^{sd}_τ, and w_τ are part of decision a_τ, variables $\delta_{\tau+1}$ and $\gamma_{\tau+1}$ are set according to constraints (3.17) and (3.18),

$$\delta_{\tau+1} = \begin{cases} \delta_\tau + 1 & \text{if } w_\tau = w_{\tau-1} \\ 1 & \text{otherwise} \end{cases} \tag{3.17}$$

$$\gamma_{\tau+1} = \begin{cases} \gamma_\tau & \text{if } w_\tau = w_{\tau-1} \\ \gamma_\tau + 1 & \text{otherwise} \end{cases} \tag{3.18}$$

and variables $\phi^{\mathrm{s}}_{\tau+1}$ and $\phi^{\mathrm{b}}_{\tau+1}$ are set with Equations (3.1) and (3.2).

Reduced costs & optimality equation. The one-period cost for period $\tau = \bar{S}_i, \ldots, E_i - 1$ is a function of w_τ, $\phi^{\mathrm{b}}_{\tau+1}$, $\phi^{\mathrm{s}}_{\tau+1}$ and the dual variable values, that is

$$c\left(w_\tau, \phi^{\mathrm{s}}_{\tau+1}, \phi^{\mathrm{b}}_{\tau+1}\right) = -\lambda^{\mathrm{w}}_{c\tau} w_\tau - \lambda^{\mathrm{wo}}_{\tau} w_\tau - \lambda^{\mathrm{b}}_{c,\tau+1}\phi^{\mathrm{b}}_{\tau+1} - \lambda^{\mathrm{s}}_{\tau+1}\phi^{\mathrm{s}}_{\tau+1} \tag{3.19}$$

The cost for being in state z_τ can be defined via a Bellman recursion as

$$v\left(z_\tau\right) = \begin{cases} \min_{z_{\tau-1} \in \mathcal{Z}_{\tau-1}(z_\tau)} \left\{ v\left(z_{\tau-1}\right) + c\left(w_{\tau-1}, \phi^{\mathrm{s}}_{\tau}, \phi^{\mathrm{b}}_{\tau}\right) \right\} & \text{for } \tau > \bar{S}_i \\ p^{\mathrm{c}}_k + p^{\mathrm{u}}_{k'} - \lambda^{\mathrm{a}}_i & \text{for } \tau = \bar{S}_i \end{cases} \tag{3.20}$$

where $\mathcal{Z}_{\tau-1}\left(z_\tau\right)$ denotes the set of all states for which a feasible decision $a_{\tau-1}$ exists that leads to state z_τ. Consider the set $\mathcal{Z}_{E_i}$ of terminal states where all bags are loaded into ULDs, i.e. z_{E_i} with $\phi^{\mathrm{b}}_{E_i} = \phi^{\mathrm{s}}_{E_i} = 0$. Pricing problem $PP^{tkk'}_{ic}$ can be solved by finding a state

$$z^* = \arg\min_{z_{E_i} \in \mathcal{Z}_{E_i}} \left\{ v\left(z_{E_i}\right) \right\}. \tag{3.21}$$

The actual profile can be derived by tracking the sequence of transitions that transforms the initial state $z_{\bar{S}_i}$ into z^*.

State-space reduction. Next, I show how the state space that is actually searched can be reduced by employing a dominance criterion. State z_τ is said to dominate another state z'_τ ($z_\tau \succ z'_\tau$), if the following conditions hold, where the variables annotated with

“ ′ ” refer to state z'_τ.

$$v(z_\tau) \leq v(z'_\tau) \tag{3.22}$$
$$\phi^{\mathrm{b}}_\tau \leq \left(\phi^{\mathrm{b}}_\tau\right)' \tag{3.23}$$
$$\phi^{\mathrm{s}}_\tau \leq \left(\phi^{\mathrm{s}}_\tau\right)' \tag{3.24}$$
$$\neg s^{\mathrm{lp}}_\tau \vee \left(s^{\mathrm{lp}}_\tau\right)' \tag{3.25}$$
$$\neg s^{\mathrm{sd}}_\tau \vee \left(s^{\mathrm{sd}}_\tau\right)' \tag{3.26}$$
$$\delta_\tau \geq \delta'_\tau \tag{3.27}$$
$$w_{\tau-1} = w'_{\tau-1} \tag{3.28}$$
$$\gamma_\tau \leq \gamma'_\tau \tag{3.29}$$

Proposition. *$z_\tau \succ z'_\tau$ implies that an optimal sequence of actions and states*

$$\left(z'_\tau, a'_\tau, z'_{\tau+1}, a'_{\tau+1}, \ldots, z'_{E_i}\right)$$

starting in state z'_τ has an objective value that cannot be better than the objective value of an optimal sequence

$$(z_\tau, a_\tau, z_{\tau+1}, a_{\tau+1}, \ldots, z_{E_i})$$

starting in state z_τ.

Proof. From (3.25) — (3.29) it follows that the set of all possible sequences $(z'_\tau, a'_\tau, z'_{\tau+1}, a'_{\tau+1}, \ldots, z'_{E_i})$ is a subset of the set of all possible sequences $(z_\tau, a_\tau, z_{\tau+1}, a_{\tau+1}, \ldots, z_{E_i})$. Because of (3.22) — (3.24) we have

$$\begin{aligned}
&\min_{(z_\tau, a_\tau, z_{\tau+1}, \ldots, z_{E_i})} \left\{ v(z_\tau) + \sum_{\tau'=\tau}^{E_i-1} c\left(w_{\tau'}, \phi^{\mathrm{s}}_{\tau'+1}, \phi^{\mathrm{b}}_{\tau'+1}\right) \right\} \\
\leq &\min_{\left(z'_\tau, a'_\tau, z'_{\tau+1}, \ldots, z'_{E_i}\right)} \left\{ v(z'_\tau) + \sum_{\tau'=\tau}^{E_i-1} c\left(w'_\tau, \left(\phi^{\mathrm{s}}_{\tau'+1}\right)', \left(\phi^{\mathrm{b}}_{\tau'+1}\right)'\right) \right\}
\end{aligned}$$

which is the desired result. □

A dominated state z_τ can be removed from further consideration, that is, any transition that starts in z_τ can be excluded. Whenever a state z_τ is reached, it is either kept, or discarded it if it is dominated by any other state that has already been evaluated. At the same time, states z'_τ that are dominated by z_τ are discarded.

Complexity. There are $n \approx \mathcal{O}\left(2 \cdot K_c^{\mathrm{w}} \cdot L \cdot J \cdot 2 \cdot \sum_\tau A_{i\tau}^{tkk'} \cdot K_c^{\mathrm{b}}\right)$ states in each period because of the variable domains (see Table (3.1)). Since there are only arcs from one point in time to the next, the network has $\mathcal{O}\left(n^2\left(E_i - \bar{S}_i\right)\right)$ arcs. Theoretically, the time complexity is $\mathcal{O}(m)$ with m as the number of arcs of the network, since each arc is evaluated once at the most. However, due to the dominance criterion, much fewer arcs need to be evaluated in practice.

3.5.4. Branch-and-price

The column generation procedure solves the linear relaxation of the MP. Hence, the solution does not necessarily satisfy the integrality conditions, and column generation must be embedded into a branch-and-bound procedure to solve the original integer problem. The overall procedure is then called *branch-and-price*. An overview can be found in Barnhart et al. (1998). Furthermore, Lübbecke and Desrosiers (2005) provide a list of research successfully applying branch-and-price.

After branching, a column may exist that would price out favorably, but is not present in the RMP. Therefore, it is necessary to continue column generation after branching (Barnhart et al., 1998). Usually, standard branching on the RMP is not a good idea because the columns that have been excluded in the RMP need to be prevented from being regenerated in the subproblems. The additional constraints may destroy the structure of the subproblems that is exploited or make them too hard to be solved in a short enough computational time. Furthermore, branching on the RMP leads to an unbalanced branch-and-bound tree (Vanderbeck, 2000). Instead, I branch on the decisions of the original integrated formulation of the problem, for which I have not presented a mathematical model, but its content should be clear from Chapter 3.4. Assume that the column generation has been solved to optimality at the current node of the branch-and-price tree n, and that there exists a flight whose solution is fractional with respect to any aspect of the decision-making: scenario, assignment to a carousel, number of workers in each period, storage depletion, or the block length or number of changes.

Assignment to carousel & scenario. If flight i is distributed to more than one carousel, flight i must be assigned to one of the carousels or not allowed to be assigned to that same carousel on the left and right branch, respectively, which partitions the solution space into two sub-spaces and excludes the current fractional solution. If flight i's planning is based on two scenarios (at most two scenarios are considered for each flight and in each stage, see Section 3.5.1), one scenario is enforced or excluded on the left and right branch, respectively. The aforementioned two branching rules are enforced by removing variables from the MP and by not solving the subproblems of the excluded carousel or scenario.

Work profile. The number of workers assigned to flight i in period $\tau = S_i, \ldots, E_i - 1$ is $w_{i\tau} = \sum_{c \in \mathcal{C}_i} \sum_{k \in \mathcal{K}^{\mathrm{c}}} \sum_{k' \in \mathcal{K}^{\mathrm{u}}} \sum_{\pi \in \mathcal{S}_{ic}^{tkk'}} x_{ic\pi}^{kk'} w_{\tau\pi}$. Let $\hat{w}_{i\tau}^{n}$ denote its value in the optimal solution at the current node n. If $\hat{w}_{i\tau}^{n}$ is fractional, $w_{i\tau} \geq \lceil \hat{w}_{i\tau}^{n} \rceil$ and $w_{i\tau} \leq \lfloor \hat{w}_{i\tau}^{n} \rfloor$ is required, at the left and right branch, respectively.

Storage depletion. The storage depletion of flight i in period $\tau = S_i, \ldots, E_i - 1$ is $s_{i\tau}^{\mathrm{sd}} = \sum_{c \in \mathcal{C}_i} \sum_{k \in \mathcal{K}^{\mathrm{c}}} \sum_{k' \in \mathcal{K}^{\mathrm{u}}} \sum_{\pi \in \mathcal{S}_{ic}^{tkk'}} x_{ic\pi}^{kk'} s_{\tau\pi}^{\mathrm{sd}}$. Let $\hat{s}_{i\tau}^{\mathrm{sd},n}$ denote its value in the optimal solution at the current node n. If $0 < \hat{s}_{i\tau}^{\mathrm{sd},n} < 1$, $s_{i\tau}^{\mathrm{sd}} = 1$ and $s_{i\tau}^{\mathrm{sd}} = 0$ is required, at the left and right branch, respectively.

Block length & number of changes. Finally, the block length and number of changes constraints need to be enforced via branching. For example, the profile π^n in Table 3.2 results adding 2/3 of π_1 and 1/3 of π_2 . Both π_1 and π_2 satisfy a block length of $L = 2$ and a maximal number of changes of $J = 1$, but π^n violates both. Assume variable $\zeta_{i\tau}$ indicates a change in the number of workers at time τ, which is fractional (given in the last row of Table 3.2). Now, three branches are created, such that $w_{i\tau} = w_{i,\tau-1}$, $w_{i\tau} < w_{i,\tau-1}$ and $w_{i\tau} > w_{i,\tau-1}$ must hold, at each branch, respectively, which requires the variable $\zeta_{i\tau}$ to be integral. For the example, requiring $w_{i,3} = w_{i,2}$ excludes profile π_1 and requiring $w_{i,3} > w_{2,\pi}$ excludes profile π_2. Requiring $w_{i,3} < w_{i,2}$ excludes both profiles. Hence, in all cases the current fractional solution is excluded, and the three branches partition the solution space into three sub-spaces. To enforce the three branching rules, all variables $x_{ic\pi}^{kk'}$ violating the corresponding branching rule are removed from the RMP, and in the subproblems decisions and states violating the rules are discarded.

Time period	1	2	3	4	5
Profile π_1 $\left(\frac{2}{3}\right)$	1	1	7	7	7
Profile π_2 $\left(\frac{1}{3}\right)$	4	4	4	1	1
Profile π^n	2	2	6	5	5
Indicator $\zeta_{i\tau}$	1	0	$\frac{2}{3}$	$\frac{1}{3}$	0

Table 3.2.: Combination of profiles

3.5.5. Search strategy

The branching rules presented are theoretically sufficient to obtain an optimal integral solution. However, the search tree becomes very large for difficult problems. Obtaining an optimal solution and proving optimality can take more than 24 hours. As the real-time environment requires a solution to be available within, say 5 minutes, the goal is to generate the best possible feasible solution within that time period. For that purpose, (i) column generation is warm started so as to improve the dual information in the first iterations and to have useful columns ready for obtaining incumbents; (ii) the branch-and-price search is guided with a *limited discrepancy search* (LDS) as proposed in Harvey and Ginsberg (1995); and (iii) incumbents are created repeatedly during the search to obtain primal bounds and to have a solution ready in case the time limit for optimization is reached.

Schedules generation. Intelligent initialization of the RMP offers speedup potential (see Vanderbeck, 2005), but is a balancing act, since RMP becomes harder to solve as more columns are added. A set of profiles is generated during preprocessing at the beginning of each stage. All profiles subject to the following restrictions determined by the following parameters are generated in a recursive algorithm, which is presented in Appendix C.1. Parameters $minDur^{\mathrm{LP}} \geq L$ and $minDur^{\mathrm{SD}} \geq L$ define the minimum duration of the loading process and the storage depletion, respectively. Parameter $slpMod$ ("slp" for start loading process and "mod" for modulo) restricts the options for the start of the LP such that it may start at time

$$\tau \in \left\{\tau = S_i, \dots, E_i - minDur^{\mathrm{LP}} \;\middle|\; slpMod \text{ divides } (E_i - \tau)\right\}.$$

In the same way, parameter $ssdMod$ restricts the options for the start of the SD. Parameter $L' \geq L$ sets the block length for the profiles generated to a value at least as large as the actual block length L. Similarly, parameter $J' \leq J$ sets the maximum number of changes to a value at most as large as the actual limit J, and parameter $\bar{G}$ defines the maximum allowed difference in numbers of workers between consecutive periods ("$\bar{G}$" for gap), i.e. $|w_\tau - w_{\tau-1}| \leq \bar{G}$. Of the generated profiles only those that are not dominated by another one are kept. A schedule π dominates a different schedule π' ($\pi \succ \pi'$) if

$$w_\tau \leq w'_\tau \qquad \forall \tau = \bar{S}_i, \ldots, E_i - 1, \tag{3.30}$$

$$\phi_{i\tau\pi}^{\mathrm{s},tkk'} \leq \left(\phi_{i\tau\pi}^{\mathrm{s},tkk'}\right)' \qquad \forall \tau = \bar{S}_i, \ldots, E_i, \tag{3.31}$$

$$\phi_{i\tau\pi}^{\mathrm{b},tkk'} \leq \left(\phi_{i\tau\pi}^{\mathrm{b},tkk'}\right)' \qquad \forall \tau = 0, \ldots, E_i. \tag{3.32}$$

Limited discrepancy search. Assume that the search strategy is a *depth first search* (DFS) in the branch-and-price tree. DFS sometimes leads directly to an optimal solution. If not, the tree is pruned, and the algorithm backtracks. However, if the search tree is deep and a node without a new incumbent in its subtree was chosen early on, a DFS searches the complete subtree, which may potentially be large, until it moves to the other branch. Limited discrepancy search (LDS) aims to overcome "wrong turns" at any depth in the search tree by deviating only a few times from the initial search path (Harvey and Ginsberg, 1995). This principle is illustrated in Figure 3.2, where the number of *allowed discrepancies* is one, and the nodes are numbered in the order they are visited. After the search returns from Node 6 to Node 5, it does not visit the branch to its right, as no further right turns are allowed because the number of right turns on the path from the root to Node 5 is equal to the number of allowed discrepancies. Therefore, the search gets back to the root node more quickly, at the cost of possibly missing a solution at the right branch below Node 5.

Solving the RMP as an integer program. Often it is possible to obtain incumbents improving the current primal bound by solving the RMP as an integer program. Especially in the first stages when $A_{i\tau}^{tkk'}$ are comparably low, we often receive an optimal solution directly for the current stage. Since solving the RMP as an IP may take a considerable amount of time, the attempt is only made at the root and, from there, every 10 nodes of

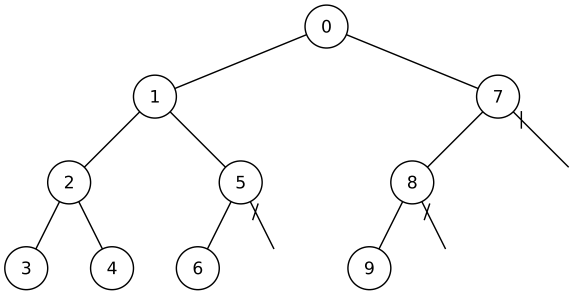


Figure 3.2.: Limited discrepancy search

the search tree, if at least 20 new columns have been generated since the last attempt. This prevents many similar problems from being solved when long cascades of branching decisions without many new columns occur. Additionally, a time limit of 100 seconds is set for the computations in each attempt, and if the time limit is reached before an optimal solution is found, an incumbent, if available, may still improve the current primal bound.

3.6. Computational Study

This section reports the experimental results of the proposed solution procedure. In order to evaluate and compare the performance of the proposed solution procedure, I developed an event-based simulation, which tracks the state of the relevant entities (flights, carousels, central storage, and ground handlers), updates the flight information, executes the planning decisions, and simulates the baggage flows. However, since modeling the BHS with its complex conveyor belt network would be too cumbersome, the BHS is treated as a black box, and backlogs of bags in the BHS do not negatively influence baggage transport. This must be kept in mind for solutions in which the carousel capacities are exceeded. In these cases, the performance of the BHS could be noticeably reduced, and as a consequence, a higher number of unloaded bags would be expected in reality. Section 3.5 laid out the proposed algorithm to solve the model in Section 3.4, and this methodology will be referred to as the *proposed procedure* or *the proposed algorithm.* The period length is set to five minutes in all experiments. The limit on the computational time is set to $\Delta = 1$ period corresponding to 5 minutes, and re-optimization is started when problem parameters change. All experiments were executed on Haswell-based nodes of the Linux cluster of the *Leibniz Supercomputing Centre of the Bavarian Academy of Sciences and Humanities*. The storage has been limited to 24 GiB and the number of cores has been limited to 14. Each core has a nominal frequency of 2.6 GHz. All of the algorithms are implemented in Java. All linear and mixed integer programs were solved using Gurobi version 8.1 and its Java API (Gurobi Optimization, 2018).

3.6.1. Instances and benchmark

The tests are based on 20 real-world problems encountered at Munich Airport. To compare the proposed procedure with the practice prevalent at most airports, all instances are solved with the greedy decision rule presented in Appendix C.2, which uses a simple logic mimicking the way a dispatcher makes a decision at the beginning of each period. While the computational times of the greedy heuristic are negligible, its performance is mediocre given the existing infrastructure at Munich Airport with many resource violations and nearly 20 mishandled bags. The proposed procedure generates feasible plans without any unloaded bags or resource violations within seconds. Therefore, in order to study challenging instances, the resource scarcity is increased. Note that resource scarcity at Munich Airport is also expected to increase given the increasing number of passengers (the number of annual passengers increased from 12 million in 1992 to more than 44 million in 2017 (Flughafen München GmbH, 2017)), and since extending the infrastructure is very costly. As the data-driven approach requires historical flight data, different degrees of resource scarcity are created by reducing the number of carousels as well as worker capacities. There are three types of carousels at Munich Airport as shown in Table 3.3. The first column ("Type ID") assigns an identifier to each type. The other columns give the carousel characteristics. There are seven carousels of type 1, 14 of type 2, and one of type 3 at Munich Airport. The layout for each set of instances is defined by a triplet, where the first, second, and third integer give the numbers of carousels of type 1, 2, and 3, respectively.

Since identical weekdays show similar air traffic patterns, I have grouped the instances by weekday, with each set of instances consisting of four instances of the same day of the week. The instance sets are listed in Table 3.4. The "Flights" and "Bags" columns show the minimum, mean, and maximum number of flights and bags, respectively. The "Carousels" column shows the number of carousels of each type, and the "Worker" column

Type ID	Working stations (K_c^{w})	Belt capacity (K_c^{b})
1	4	20
2	4	25
3	6	40

Table 3.3.: Carousel types

ID	Flights			Bags			Carousels			Worker
	min	avg	max	min	avg	max	1	2	3	(K_τ^{wo})
Mon	86	89	92	5,715	6,270	7,012	6	7	1	70
Tue	85	97	105	4,264	5,380	6,929	5	7	1	42
Wed	91	95	98	4,464	5,048	5,527	5	6	1	38
Thu	98	102	107	5,160	5,706	6,525	5	5	1	36
Fri	414	425	436	20,509	21,324	22,138	5	5	1	40

Table 3.4.: Instances

shows the number of workers. In contrast to all other instances, the Friday instances cover not only one peak period but the complete planning day, which generally has three peaks during the day at Munich Airport. On average, a worker requires 30 seconds to load a bag, which gives a loading rate of $r^{\mathrm{l}} = 10$ bags per period. The storage depletion rate is set to $r^{\mathrm{s}} = 19$ bags per period, and the storage capacity is $K^{\mathrm{s}} = 6,000$ bags.

For each departing flight, there are on average 12 observed check-in arrival processes available, that is, 12 times the same flight number and the same destination on the same day of the week departing from Munich Airport. For a given day of the week, planning days have been selected randomly, which provide the actual data used in the simulation, while the remaining observations are used in the optimization. For the transfer arrival processes, the quantity of baggage an inbound flight delivers for each outbound flight becomes known when baggage handling at the origin airport is finished.

Instances	Greedy				Main				
	Left bags		Belt vios		Comp. time	Left bags		Belt vios	
	avg	max	avg	max	avg mins	avg	max	avg	max
Mon	816	989	117	132	0.5	0.3	1	0.3	1
Tue	426	786	73	103	0.4	0.5	2	0.0	0
Wed	432	586	52	58	0.5	3.0	10	0.0	0
Thu	687	927	76	101	0.8	74.7	201	10.7	36
Fri	1,805	1,992	183	203	1.9	4.0	5	6.0	7

Table 3.5.: Results for baseline setup

3.6.2. Baseline setup

At first, the parameters determining the worker flexibility are set to values reflecting practice at Munich Airport. The block length is set to 30 minutes ($L = 6$), which also means that the duration of the loading process must be at least 30 minutes. The number of changes is restricted to $J = 2$, which means that apart from the start and end of a flight's LP, the number of assigned workers can be changed twice. Table 3.5 shows the results. Each row corresponds to a set of instances for a weekday. The columns contain performance measures for both the greedy heuristic and the proposed algorithm, and the computation times (average minutes per re-optimization) for the proposed algorithm. The "Left bags" columns report the number of bags that are not loaded into ULDs by the end of the baggage handling process. The extent to which the conveyor belt capacity is violated is measured as follows. One *belt capacity violation* of a carousel is one period in which one or more bags must remain in the BHS because the carousel's belt capacity is completely used up. The *total belt capacity violations* are the sum of the belt capacity violations over all carousels, which are reported in the "Belt vios" columns. No violations of the storage capacity were observed in the simulation. The table shows that the greedy heuristic is outperformed in all cases, with several hundred unloaded bags, on average, compared to only few left bags and capacity violations with the proposed algorithm. The results are worse for the Thursday instances, which have the lowest capacity with only 11 carousels in total and 36 workers. Therefore, the Thursday instances can be considered a stress test and will be used in the subsequent experiment, in which I analyze worker flexibility.

Flexibility		Greedy		Main		
Block length (L)	Max. no. of changes (J)	Left bags	Belt vios	Comp. time (avg mins)	Left bags	Belt vios
3	∞	62	175	1.3	2 (1%)	0 (0%)
3	2	168	137	1.4	2 (1%)	1 (3%)
3	0	324	123	0.8	2 (1%)	1 (3%)
6	∞	441	164	0.8	133 (66%)	24 (67%)
6	**2**	479	146	0.8	201 (100%)	36 (100%)
6	0	729	117	0.8	250 (124%)	42 (117%)
12	∞	1250	109	0.8	1,812 (901%)	52 (144%)
12	2	1250	109	0.8	1,943 (967%)	55 (153%)
12	0	1478	101	0.5	2,176 (1083%)	50 (139%)

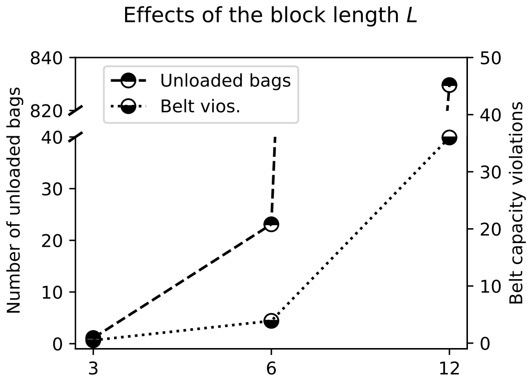
Table 3.6.: Computational results for a difficult instance

3.6.3. Evaluation of flexibility

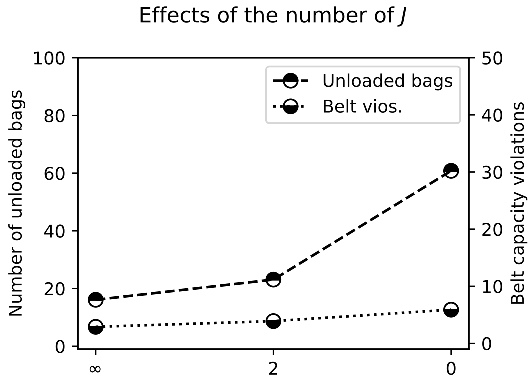
Here, I will consider different degrees of worker flexibility to gain insights into the value thereof. To that end, I vary the block length L and the number of times the number of workers is allowed to change during a flight's loading process J. All instances are solved with $(L, J) = (L, \infty), (L, 2)$ and $(L, 0)$ for $L = 3, 6, 12$ periods corresponding to 15, 30 and 60 minutes, respectively. $J = 0$ requires the number of workers to be constant throughout a flight's loading process. The effects of flexibility are first illustrated by looking at the results for the most difficult instance, in terms of the number of left bags. The results are shown in Table 3.6, in which each row corresponds to an experiment with different values for the parameters that determine the flexibility, and the first row is the most flexible with a block length of 15 minutes and an arbitrary number of changes $(L, J) = (3, \infty)$, and the last row is the least flexible with a block length of one hour and rectangular work profiles $(L, J) = (12, 0)$. The left bags and belt capacity violations of our algorithm are also given as percentages of the baseline setup in parentheses behind the value.

The proposed procedure outperforms the greedy heuristic, except when the flexibility is very low. When planning with the greedy heuristic, flexibility has a strong effect on the number of mishandled bags, but there is no clear relationship between flexibility and the capacity violations. When planning with the proposed procedure, flexibility positively effects both the number of mishandled bags and the capacity violations. With a block length of 15 minutes and an arbitrary number of changes, there are only two mishandled bags and no capacity violations. Keeping the block length at that value and reducing J leads to two capacity violations. The good performance for $J = 0$ seems surprising at first. Taking a closer look at the solutions, it is observed that in general there are three types of profiles: (i) long loading process with relatively few workers and early storage depletion, (ii) short loading process with late storage depletion with many workers, and (iii) the same as (ii), but with only one worker. In the case of (i), bags are sent directly to the carousel over a long period. In the case of (ii), bags are buffered in the storage and then sent to the carousel to be loaded into ULDs by multiple workers in a short period of time, and case (iii) occurs for flights with very little baggage, where a single worker can load all bags within 15 minutes. If the block length was longer, a worker would need to be assigned for at least that long, thereby wasting capacity that otherwise could be used for other flights. This is precisely what happens when the block length is set to 30 minutes ($L = 6$) or one hour ($L = 12$). With $L = 6$, the wasted capacity leads to more than 100 unloaded bags. This shows that being able to change the number of workers allows the loading processes to be matched to fluctuating baggage streams more efficiently, and thus leads to a higher resource utilization. When the block length is set to one hour ($L = 12$), the utilization of the workers assigned to the "small flights" is very low on average, and the capacity that is left for the remaining flights is insufficient. As a result, the procedure is forced to work with too low scenarios, and we obtain results that are worse than the greedy heuristic. Hence, for functioning baggage handling under scarce resources, flexibility is key.

Table C.1 of Appendix C.3 shows the aggregated results of all instances. With the most flexible setting $(L, J) = (3, \infty)$ and the proposed procedure, the maximum number of unloaded bags and capacity violations is 3 and 8, respectively. Hence, a high degree of worker flexibility allows even the most difficult instances to be solved with good results. As illustrated in Figure 3.3a, the block length has a strong effect on the number of unloaded bags, as well as the belt capacity violations. With a block length of $L = 12$, more than 800 bags cannot be loaded on average. An explanation is that scarce resources are assigned to small flights longer than necessary, and hence, are wasted. In the unlikely event that it is impossible to assign workers and working stations to a flight for the minimum duration, the baggage handling process of that flight cannot start, and all its bags must be delivered



(a) Varying L



(b) Varying J

Figure 3.3.: Varying flexibility parameters

by a later airplane. The effect of the number of allowed changes J is illustrated in Figure 3.3b. The average number of unloaded bags decreases from 60.8 when only a constant number of workers is allowed for each flight ($J = 0$) to 16.1 when the number of changes is unrestricted ($J = \infty$), which corresponds to a decrease of 74%.

3.7. Conclusions

In this chapter, I presented a new model for dynamic baggage handling with flexible work profiles. A flight's work profile determines how many workers should work over the course of the baggage handling period. Solutions to the model provide the assignments of flights to baggage carousels, the start of the baggage loading process, a work profile as well as the start of the storage depletion for each flight. The objective is to generate robust plans so as to make sure bags can be loaded in time, i.e. before the end of a flight's baggage handling process, and avoid backlogs in the baggage handling system. To obtain solutions to the model in as close to real-time as possible, I developed a column-generation-based heuristic. The hierarchic objective allows the problem to be solved in stages. At the beginning of each stage, the restricted master problem is fed with a set of predefined columns. To obtain integral solutions, column generation is embedded into branch-and-price procedure. Incumbents are obtained by solving the master problem as an integer program repeatedly during the search for an integral solution. Using this combined methodology, it is possible to solve real-world instances and generate high-quality solutions even for problems with very scarce resources.

The modeling approach allows for the desired degree of worker flexibility to be adjusted, and several managerial implications can be taken from the computational study. First, the proposed procedure outperforms the greedy heuristic, which mimics manual decision-making by a dispatcher. Second, the inflow of bags to the carousel can be controlled, to some extent, by scheduling the start of the loading process and storage depletion. However, to match the total loading rate to the flow of incoming bags more effectively, flexible work profiles must be employed. It is crucial to not require the number of assigned workers to be constant for a long time. For small flights, luggage can often be loaded by a single worker within a short time, and in such cases assigning a worker for longer than necessary is a waste of capacity that could be utilized for other flights. Additionally, when capacity is scarce, planning with rectangular work profiles, i.e. constant numbers of workers throughout the loading process for each flight, leads to poor solutions.

With respect to future research, several opportunities exist for expanding the scope of the problem, for example by taking into account the shifts, breaks, and movements of individual workers. Another possibility would be to integrate the personnel planning of the ground handling companies. Furthermore, with regard to alternative solution methodologies, heuristics with very short computational times may be worthwhile from a practical point of view.

4. Conclusion

In this dissertation two planning problems related to the field of workforce planning were presented. While the workforce capacity planning problem of Chapter 2 belongs to the class of strategic planning problems, the baggage handling problem of Chapter 3 belongs to the class of operative planning problems. While the former establishes and maintains a permanent workforce, the latter, assuming specific worker capacities, simultaneously optimizes task schedules and worker assignments. These worker capacities result from the rostering of the ground handling company contracted by the corresponding airline. In both problems, uncertainty is explicitly considered in the models, but in different ways. In the strategic workforce capacity planning, uncertainty is represented by resignation probabilities, which need to be estimated based on historical data, and the problem is modeled as a Markov decision problem. In the baggage handling problem, uncertainty is dealt with by integrating historical flight data into the integer programming model without the need to estimate a point estimate or a distribution. Both model formulations are computationally intractable with standard algorithms, and therefore, it was necessary to design customized solution procedures for both cases. To solve realistic instances of the workforce capacity planning problem in Chapter 2, I developed three decision rules. The first rule was inspired by inventory control literature, and I extended that rule twice to obtain improved results. To determine the decision rule's parameters, a search procedure was developed. This combined methodology allows for solving realistic problems, while the standard algorithms fail already for medium sized problems. To solve realistic instances of the baggage handling problem in Chapter 3, I developed a column-generation-based heuristic. Certain features of the procedure allow for solving the problem in a limited period as required for the real-time environment: first, column generation is warm started by adding promising work profiles generated during preprocessing; second, the master problem is solved repeatedly during the search for an integral solution to obtain incumbents; third, the search tree is truncated according to the limited discrepancy search methodology; and fourth, computational time is limited. Using this methodology, high-quality solutions can be generated for real-world problems – even with very scarce resources.

In summary, the research in this dissertation demonstrates how specific workforce planning problems incorporating uncertainty can be solved with highly customized solution procedures. A relevant consideration from a practitioners point of view is to keep the complexity as low as possible, so as to keep development cycles short, and to improve maintainability and comprehensibility. The latter can contribute to the acceptance at the management and the user level. However, at the moment, no general purpose algorithm is powerful enough to solve such challenging optimization problems under uncertainty. With respect to Chapter 2, a useful path for future research would be to extend the scope of the model by allowing more complex skill structures. Another way to extend the scope would be to allow limited hiring of skilled employees. Furthermore, it is sometimes possible to conduct training in different modes, that is, either slowly while the trainee is relatively productive, or quickly while the trainee is completely removed from the shop floor. When such an opportunity exists, the question arises whether or not it is better to release the worker for the shorter period. Another direction for further research would be to modify the one-period cost function in order to integrate aspects of the lower level planning, i.e. the rostering. With respect to Chapter 3, an interesting path for future research would be to include worker movements between baggage handling facilities as well as their shifts and breaks. Similarly, integrating the personnel scheduling of the ground handling companies could reduce personnel costs by better matching the staffing levels to the actual workload. The scope of the problem could further be extended to include gate assignments, such that the distances between baggage carousels and the flights' parking positions are minimized. Furthermore, optimizing the baggage transport systems, would be an approach to further improve the system performance. Whether in the manufacturing or the service industries, including uncertainty increases the difficulties, yet produces more realistic models and better planning results.

A. Notation Workforce Planning

Table A.1.: Notation used in the model

Sets, parameters, functions	
$\mathcal{T} = \{0, \dots, \infty\}$	planning horizon
L	number of proper skill levels
$\mathcal{L} = \{1, 2, \dots, L\}$	proper skill levels
0	dummy skill level representing an external location
$\mathcal{L}_0 = \{0, 1, \dots, L-1\}$	skill levels from which workers can start training
Δ_l	duration for training from level $l \in \mathcal{L}_0$ to $l+1$
$\mathcal{T}_{tl} = \{t - \Delta_l + 1, \dots, t-1\}$	training start times for training starting at level $l \in \mathcal{L}_0$ relative to current time $t \in \mathcal{T}$
Π	set of policies
$A^{\pi}(z_t)$	decision rule of policy $\pi \in \Pi$ as a function of the pre-decision state z_t at time $t \in \mathcal{T}$
γ	discounting factor
$C(z_t^{\mathrm{x}}) = C^{\mathrm{wf}}(z_t^{\mathrm{x}}) + C^{\text{-}}(z_t^{\mathrm{x}})$	one-period costs as a function of post-decision state z_t^{x} at time $t \in \mathcal{T}$
$C^{\mathrm{wf}}(z_t^{\mathrm{x}})$	one-period workforce costs as a function of post-decision state z_t^{x} at time $t \in \mathcal{T}$
$C^{-}(z_t^{\mathrm{x}})$	one-period under-staffing penalty costs as a function of post-decision state z_t^{x} at time $t \in \mathcal{T}$
c_l^{t}	costs for a level l worker in training ($l \in \mathcal{L}_0$)
c_l^{s}	salary for a level l worker ($l \in \mathcal{L}$)
D_l	minimum number of required level l workers ($l \in \mathcal{L}$)
c^{-}	penalty costs per unit of missing capacity per period

ρ_l^{c}	productivity of a worker in training on current level $l = 1, \ldots, L-1$
ρ_l^{t}	productivity of a level l worker in training on target level $l+1$ ($l \in \mathcal{L}_0$)
$K_l\left(z_t^{\mathrm{x}}\right)$	capacity at level $l \in \mathcal{L}$ as a function of the post-decision state z_t^{x} at time $t \in \mathcal{T}$
K	incremental capacity (local variable in Algorithm 1)
Random variables and probabilities	
p_l^{r}	probability that a level l worker resigns ($l \in \mathcal{L}$)
$X_{tl}^{\mathrm{r}} \sim B\left(z_{t-1,l}^{\mathrm{ax}}, p_l^{\mathrm{r}}\right)$	binomially distributed random variable for the number of level l workers ($l \in \mathcal{L}$) that hand in their resignation in period $(t-1, t)$ to resign at time $t \in \mathcal{T} \setminus \{0\}$
$X_t^{\mathrm{r}} = \left(X_{tl}^{\mathrm{r}}\right)$	all random variables for resignation at time $t \in \mathcal{T} \setminus \{0\}$
State variables	
$\mathcal{Z}$	pre-decision state space
z_{tl}^{a}	number of available workers at skill level $l \in \mathcal{L}$ at time $t \in \mathcal{T}$, before a decision is made
$z_{tlt'}^{\mathrm{t}}$	number of workers at skill level $l \in \mathcal{L}_0$ in training for level $l+1$ at time $t \in \mathcal{T}$, before a decision is made, that started their training at $t' \in \mathcal{T}_{tl}$
z_t	pre-decision state at time $t \in \mathcal{T}$
z_{tl}^{ax}	number of available workers at skill level $l \in \mathcal{L}$ at time $t \in \mathcal{T}$, after a decision has been made
$z_{tlt'}^{\mathrm{tx}}$	number of workers at skill level $l \in \mathcal{L}_0$ in training for level $l+1$, that started their training at $t' \in \{t - \Delta_l + 1, \ldots, t\}$, at time $t \in \mathcal{T}$, after a decision has been made
z_t^{x}	post-decision state at time $t \in \mathcal{T}$
z_{tl}^{tx}	number of workers in training at skill level $l \in \mathcal{L}_0$ at time $t \in \mathcal{T}$ post-decision
Decision variables	

$x_{tl} \in \mathbb{N}_0$	Number of workers who start training at level $l \in \mathcal{L}_0$ at time $t \in \mathcal{T}$
$x_t = (x_{tl})_{l \in \mathcal{L}_0}$	Decisions vector at time $t \in \mathcal{T}$
System model	
$S^{\mathrm{Mx}}(z_t, x_t) = z_t^{\mathrm{x}}$	transition from pre-decision state z_t at time $t \in \mathcal{T}$ to post-decision state z_t^{x}
$S^{\mathrm{MX}}\left(z_t^{\mathrm{x}}, X_{t+1}^{\mathrm{r}}\right) = z_{t+1}$	transition from post-decision state z_t^{x} at time $t \in \mathcal{T}$ to next pre-decision state z_{t+1}
$S^{\mathrm{M}}\left(z_t, x_t, X_{t+1}^{\mathrm{r}}\right) = z_{t+1}$	transition from pre-decision state z_t at time $t \in \mathcal{T}$ to pre-decision state z_{t+1}

Table A.2.: Notation used in the solution methodology

Notation for defining policies	
S_l	base-stock for level $l \in \mathcal{L}$
$S = (S_1, \ldots, S_L)$	base-stock vector
M_l	minimum-stock for level $l = 1, \ldots, L-1$
$M = (M_1, \ldots, M_{L-1})$	minimum-stock vector
C_l	training-capacity for level $l \in \mathcal{L}_0$
$C = (C_l)_{l \in \{l \in \mathcal{L}_0 : \rho_l^{\mathrm{t}} < 0\}}$	training-capacity vector
z_{tl}	number of workers that are at level $l \in \mathcal{L}$ or in training to become level l at time $t \in \mathcal{T}$
$n_{tl} = \sum_{k=l}^{L} S_k - z_{tl} \in \mathbb{Z}$	shortage ($n_{tl} > 0$) or surplus ($n_{tl} < 0$) at level $l \in \mathcal{L}$ at time $t \in \mathcal{T}$
$[a]^+$	shorthand for $\max\{0, a\}$
$A(S, z_t) \mapsto x_t$	base-stock decision rule that returns a decision vector x_t given base-stock vector S and pre-decision state z_t at time $t \in \mathcal{T}$
$A(S, M, z_t) \mapsto x_t$	base/minimum-stock decision rule that returns a decision vector x_t given base-stock vector S, minimum-stock vector M and pre-decision state z_t at time $t \in \mathcal{T}$

$A(S, M, C, z_t) \mapsto x_t$	base/minimum-stock/training-capacity decision rule that returns a decision vector x_t given base-stock vector S, minimum-stock vector M, training-capacity vector C and pre-decision state z_t at time $t \in \mathcal{T}$
Notation for policy search	
$\hat{C}_t^{\pi}$	observed one-period costs for policy $\pi \in \Pi$ in period $(t, t+1)$ $(t \in \mathcal{T})$
$\hat{X}_t^{\mathrm{r}}$	observed resignations during period $(t-1, t)$ $(t \in \mathcal{T} \setminus \{0\})$
$\hat{z}_t$	observed state at time t $(t \in \mathcal{T} \setminus \{0\})$
$\bar{C}_n^{\pi}$	sample mean of $\hat{C}_t^{\pi}$ after n observations
n^{π}	sample size for policy $\pi \in \Pi$
$M_n^{2,\pi}$	sum of squares of differences for policy $\pi \in \Pi$ after n observations
$S_n^{2,\pi}$	sample variance for policy $\pi \in \Pi$ after n observations
S_n^{π}	sample standard deviation for policy $\pi \in \Pi$ after n observations
V	hash map with policies as keys and triplets with sample statistics $\left(n_{\pi}, \bar{C}_n^{\pi}, M_n^{2,\pi}\right)$ as values
$N(\pi)$	neighborhood of $\pi \in \Pi$
$\alpha(\pi, \phi)$	dynamic significance level of test comparing π and $\phi \in \Pi$
obs	threshold on number of observations for policy comparison

B. Notation Baggage Handling

Table B.1.: Notation

Notation	
$\mathcal{T} = \{0, \dots, T\}$	Planning horizon of the complete day
Δ	Time limit in number of periods for re-optimization
P^t	(Re-)optimization problem that is solved starting at time $t = -\Delta, \dots, T - \Delta - 1$
$t'_t = t + \Delta$	Time when the solution of optimization problem P^t takes effect for $t = 0 - \Delta, \dots, T - 1 - \Delta$
Δ^{h}	Number of periods to limit the planning horizon
$\mathcal{T}_t = \left\{t'_t, \dots, t'_t + \Delta^{\mathrm{h}}\right\}$	Planning horizon of problem P^t for $t = 0, \dots, T - 1$
Carousels	
$\mathcal{C}$	Set of carousels
K^{b}_c	Conveyor belt capacity in number of bags of carousel $c \in \mathcal{C}$
K^{w}_c	Number of working stations of carousel $c \in \mathcal{C}$
$\mathcal{C}_i$	Carousels that can be used for flight $i \in \mathcal{F}$
Storage	
K^{s}	Storage capacity
r^{s}	Storage depletion rate per period
Workers	
K^{wo}_τ	Number of available workers at time $\tau \in \mathcal{T}$
r^{l}	Loading rate per worker (and working station) in bags per period
Flights	
$\mathcal{F}$	Set of flights

$\mathcal{F}^t$	Flights that are included in problem P^t
E_i	End of baggage handling of flight $i \in \mathcal{F}$
$\mathcal{K}^{\mathrm{c}} = \{1, \ldots, K^{\mathrm{c}}\}$	Index set for certain bags
$\mathcal{K}^{\mathrm{u}} = \{0, \ldots, K^{\mathrm{u}}\}$	Index set for uncertain baggage
$\mathcal{K}^{\mathrm{c}} \times \mathcal{K}^{\mathrm{u}}$	Set of scenarios
$(k, k') \in \mathcal{K}^{\mathrm{c}} \times \mathcal{K}^{\mathrm{u}}$	Scenario
$A_{i\tau}^{tkk'}$	Number of bags arriving for flight $i \in \mathcal{F}$ in period $\tau = -1, \ldots, E_i - 1$ according to scenario $(k, k') \in \mathcal{K}^{\mathrm{c}} \times \mathcal{K}^{\mathrm{u}}$ and the information available at time $t = -\Delta, \ldots, T - \Delta - 1$
$\phi_{i\tau\pi}^{\mathrm{b},tkk'}$	Number of bags on the conveyor belt for flight $i \in \mathcal{F}$ at time $\tau = S_i, \ldots, E_i$ with work profile $\pi \in \mathcal{S}_i$ according to scenario $(k, k') \in \mathcal{K}^{\mathrm{c}} \times \mathcal{K}^{\mathrm{u}}$ and the information available at time $t = -\Delta, \ldots, T - \Delta - 1$
$\phi_{i\tau\pi}^{\mathrm{s},tkk'}$	Number of bags in storage for flight $i \in \mathcal{F}$ at time $\tau = 0, \ldots, E_i$ with work profile $\pi \in \mathcal{S}_i$ according to scenario $(k, k') \in \mathcal{K}^{\mathrm{c}} \times \mathcal{K}^{\mathrm{u}}$ and the information available at time $t = -\Delta, \ldots, T - \Delta - 1$
LP / SD	Loading process / Storage depletion
S_i	Earliest start of the LP of flight $i \in \mathcal{F}$
s_i^{slp} / s_i^{ssd}	Start of the LP / SD of flight $i \in \mathcal{F}$
$s_{i\tau}^{\mathrm{lp}}$ / $s_{i\tau}^{\mathrm{sd}}$	$= 1$ if the LP / SD of flight $i \in \mathcal{F}$ is in progress in period $\tau = S_i, \ldots, E_i - 1$
$w_{i\tau}$	Number of workers assigned to flight $i \in \mathcal{F}$ in period $\tau = S_i, \ldots, E_i - 1$
L	Block length
J	Number of times the number of workers is allowed to change during a flight's LP
π	Schedule and work profile tuple
$\mathcal{S}_{ic}^{tkk'}$	Set of profiles π for flight $i \in \mathcal{F}^t$ and carousel $c \in \mathcal{C}_i$ based on $\left(A_{i\tau}^{tkk'}\right)_{\tau=-1,\ldots,E_i-1}$
$\mathcal{S}_{ic\tau}^{tkk'}$	Subset of $\mathcal{S}_{ic}^{tkk'}$ where the LP is in progress during period $\tau = S_i, \ldots, E_i - 1$

$\tilde{\mathcal{S}}_{ic}^{tkk'}$	Subset of $\mathcal{S}_{ic}^{tkk'}$ of the restricted master problem
Decision variables	
$x_{ic\pi}^{kk'}$	1, if flight $i \in \mathcal{F}^t$ is assigned to carousel $c \in \mathcal{C}_i$ with scenario $(k, k') \in \mathcal{K}^{\mathrm{c}} \times \mathcal{K}^{\mathrm{u}}$ and profile $\pi \in \mathcal{S}_{ic}^{tkk'}$; 0 otherwise

C. Supplemental Material Baggage Handling

C.1. Profile Generation

The schedules complying with the rules introduced in Section 3.5.5 are generated as follows. For each duration of the loading process d that is allowed according to $slpMod$, a set of work profiles is generated. For a given duration of the loading process d, the work profiles are built recursively as shown in Algorithm 4. The recursion progresses through the time periods until the numbers of workers have been set for all d periods. Buffer $(w_0, \ldots, w_{d-1})$ stores an incomplete work profile. Each time a work profile is completed, it is added to the list of results $\mathcal{P}$. Parameter c tracks the number of changes of workers. At the start of each recursion, if values are set for all d periods, the profile in the buffer is added to the results, and the recursion ends. Otherwise, if no more changes are allowed because of parameter J' or further changes would violate the block length L', the remaining values are set equal to the previous period, and the profile is added to the results. Otherwise, the algorithm loops over the values allowed according to the maximum gap $\bar{G}$. In each iteration, if the new value is equal to the previous value, the new value is stored in the buffer, and the recursion continues with the next period without incrementing the number of changes c. If the value differs from the previous one, the value is set for the next L' periods and the recursion continues L' periods later with the number of changes incremented by one.

Next, each work profile is combined with each allowed start of the storage depletion, completing the schedule. The storage and load on the carousel are determined with equations (3.1) and (3.2). If both storage and belt are empty at the end of the LP, the schedule π is a candidate to be added to $\tilde{\mathcal{S}}_{ic}^{tkk'}$.

Algorithmus 4 : Profile generation

Function *generateProfiles*

Input : current period t,
profile buffer $(0, w_0, \ldots, w_{d-1})$,
changes c,
set of profiles $\mathcal{P}$

if $t = d$ **then** $\mathcal{P} \leftarrow \mathcal{P} \cup \{(w_0, \ldots, w_{d-1})\}$, **return**
if $c \geq J' \;\vee\; d - t < L'$ **then**
  while $t < d$ **do** $w_t \leftarrow w_{t-1}$, $t \leftarrow t + 1$
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{(w_0, \ldots, w_{d-1})\}$, **return**
for $w = \max\left\{1, w_{t-1} - \bar{G}\right\}, \ldots, \min\left\{\max_{c \in \mathcal{C}_i} K_c^w, w_{t-1} + \bar{G}\right\}$ **do**
  if $w = w_{t-1}$ **then**
    $w_t \leftarrow w$, generateProfiles$(t + 1, c, (w_0, \ldots, w_{d-1}))$
  else
    for $t' = t, \ldots, \min\{t + L', d\} - 1$ **do** $w_{t'} \leftarrow w_{t-1}$
    generateProfiles$(t' + 1, c + 1, (w_0, \ldots, w_{d-1}))$

C.2. Greedy Decision Rule

The greedy heuristic mimicking the decision-making of a human dispatcher is outlined as follows. In general, a human dispatcher distributes the available resources (workers and working stations) to flights to ensure that "big flights" with a lot of baggage receive more resources than "small flights". Therefore, to automatize this process, (i) a measure for the size of a flight or the resources needed and (ii) a way to distribute resources depending on that measure are required, so that both (i) and (ii) correspond to the logic a dispatcher uses. At each decision, the flights that cannot be changed due to the block length restriction or due to the limit on the number of changes receive the same number of workers and thus working stations as previously. The remaining flights are added to a candidate list and sorted in decreasing order by the expected number of workers required per period. More precisely, by

$$WS_i \coloneqq \frac{1}{(E_i - t)\, r^1} \left(\left(\phi_t^{\mathrm{s}} + \phi_t^{\mathrm{b}}\right) + \sum_{\tau = t}^{E_i - 1} \bar{A}_{i\tau} \right),$$

where $\phi_t^{\mathrm{s}} + \phi_t^{\mathrm{b}}$ are the bags already in the system (on the carousel or in the storage) at time t, and $\bar{A}_{i\tau}$ denotes the expected number of bags arriving for flight i in period

$\tau = t, \dots, E_i - 1$, which is derived from past observations. Note that WS_i is not rounded, which leads to very few ties. Next, the following steps are executed iteratively, where WS_i will be decremented each time flight i receives a worker and working station.

Step 0 If the candidate list is empty or if there are no resources left, stop and return the solution.

Step 1 Take the first flight from the list. Let this flight be i^*.

Step 2 If flight i^* is in progress, check whether the number of workers at the assigned carousel can be increased by one. If that is impossible because there are no workers left or because the carousel has no additional carousels, remove the flight from the candidate list, and go to Step 0. Otherwise, increase the number of workers of flight i^* by one and decrease WS_{i^*} by one because the flight has just received a worker. Sort the candidate list in decreasing order by WS_i, and go to Step 0.

Step 3 If flight i^* is not in progress, check whether resources are available to start the loading process with one worker. If not, remove the flight from the candidate list and go to Step 0. Otherwise, start the LP with one worker at the carousel with the most free working stations and start the SD, decrease WS_{i^*} by one, set the flight to be in progress, sort the candidate list in decreasing order by WS_i, and go to Step 0.

C.3. Aggregated Results

Table C.1 shows the aggregated results of all 20 instances. Each row corresponds to different values for the parameters that determine the flexibility.

Flexibility		Greedy				Main				
Block	Max. no. of	Left bags		Belt vios		Comp. time	Left bags		Belt vios	
length (L)	changes (J)	avg	max	avg	max	avg secs	avg	max	avg	max
3	∞	55	274	95	201	41	0	3	0	8
3	2	102	300	94	174	39	1	19	1	7
3	0	308	727	102	223	33	1	10	1	15
6	∞	277	994	93	168	39	16	173	3	30
6	2	298	1,035	89	169	42	23	211	4	45
6	0	1,010	2,122	90	190	33	61	659	6	76
12	∞	800	2,675	73	149	31	818	6,296	38	295
12	2	800	2,675	73	149	34	830	5,970	36	281
12	0	1,467	2,969	82	180	35	851	6,418	46	323

Table C.1.: Aggregated computational results

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die bei der promotionsführenden Einrichtung Fakultät für Wirtschaftswissenschaften der TUM zur Promotionsprüfung vorgelegte Arbeit mit dem Titel “Workforce Planning in Manufacturing and Service Industries” unter der Anleitung und Betreuung durch Prof. Dr. Rainer Kolisch ohne sonstige Hilfe erstellt und bei der Abfassung nur die gemäß § 6 Ab. 6 und 7 Satz 2 angebotenen Hilfsmittel benutzt habe. Ich habe keine Organisation eingeschaltet, die gegen Entgelt Betreuerinnen und Betreuer für die Anfertigung von Dissertationen sucht, oder die mir obliegenden Pflichten hinsichtlich der Prüfungsleistungen für mich ganz oder teilweise erledigt. Ich habe die Dissertation in dieser oder ähnlicher Form in keinem anderen Prüfungsverfahren als Prüfungsleistung vorgelegt. Ich habe den angestrebten Doktorgrad noch nicht erworben und bin nicht in einem früheren Promotionsverfahren für den angestrebten Doktorgrad endgültig gescheitert. Die öffentlich zugängliche Promotionsordnung der TUM ist mir bekannt, insbesondere habe ich die Bedeutung von § 28 (Nichtigkeit der Promotion) und § 29 (Entzug des Doktorgrades) zur Kenntnis genommen. Ich bin mir der Konsequenzen einer falschen Eidesstattlichen Erklärung bewusst. Mit der Aufnahme meiner personenbezogenen Daten in die Alumni-Datei bei der TUM bin ich einverstanden.

Ort, Datum, Unterschrift

Bibliography

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

Ahn, H.-S., Righter, R., and Shanthikumar, J. G. (2005). Staffing decisions for heterogeneous workers with turnover. *Mathematical Methods of Operations Research*, 62(3):499–514.

Al-Yakoob, S. M. and Sherali, H. D. (2007). Multiple shift scheduling of hierarchical workforce with multiple work centers. *Informatica*, 18(3):325–342.

Alfares, H. K. (2004). Survey, categorization, and comparison of recent tour scheduling literature. *Annals of Operations Research*, 127(1):145–175.

Anderson, E. G. J. (2001). The nonstationary staff-planning problem with business cycle and learning effects. *Management Science*, 47(6):817–832.

Anthony, R. N. (1965). *Planning and Control Systems: A Framework for Analysis*. Planning and Control Systems: A Framework for Analysis. Division of Research, Graduate School of Business Administration, Harvard University.

Ascó, A., Atkin, J. A. D., and Burke, E. K. (2014). An analysis of constructive algorithms for the airport baggage sorting station assignment problem. *Journal of Scheduling*, 17(6):601–619.

Bailey, C. D. (1989). Forgetting and the learning curve: A laboratory study. *Management Science*, 35(3):340–352.

Baker, K. R. (1976). Workforce allocation in cyclical scheduling problems: A survey. *Journal of the Operational Research Society*, 27(1):155–167.

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

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

Ball, M., Barnhart, C., Nemhauser, G., and Odoni, A. (2007). Chapter 1 air transportation: Irregular operations and control. In Barnhart, C. and Laporte, G., editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, pages 1 – 67. Elsevier.

Barbot, C. (2012). Opening ground handling markets to competition: Effects on welfare. *Transportation Science*, 46(4):536–546.

Bard, J. F. (2004). Staff scheduling in high volume service facilities with downgrading. *IIE Transactions on Scheduling & Logistics*, 36(10):985–997.

Bard, J. F. and Purnomo, H. W. (2005). A column generation-based approach to solve the preference scheduling problem for nurses with downgrading. *Socio-Economic Planning Sciences*, 39(3):193–213.

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

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., and Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329.

Beasley, J. E., Krishnamoorthy, M., Sharaiha, Y. M., and Abramson, D. (2000). Scheduling aircraft landings - the static case. *Transportation Science*, 34(2):180–197.

Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations Research*, 52(1):35 – 53.

Billionnet, A. (1999). Integer programming to schedule a hierarchical workforce with variable demands. *European Journal of Operational Research*, 114(1):105–114.

Bolat, A. (2000). Procedures for providing robust gate assignments for arriving aircrafts. *European Journal of Operational Research*, 120(1):63 – 80.

Bordoloi, S. K. and Matsuo, H. (2001). Human resource planning in knowledge-intensive operations: A model for learning with stochastic turnover. *European Journal of Operational Research*, 130(1):169–189.

Bruecker, P. D., den Bergh, J. V., Beliën, J., and Demeulemeester, E. (2015). Workforce planning incorporating skills: State of the art. *European Journal of Operational Research*, 243(1):1–16.

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

Bureau of Transportation Statistics (BTS) (2018). On-Time Performance - Flight Delays at a Glance.

Burke, E. K., De Causmaecker, P., Berghe, G. V., and Van Landeghem, H. (2004). The state of the art of nurse rostering. *Journal of Scheduling*, 7(6):441–499.

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

Chvátal, V. (1983). *Linear Programming*. W. H. Freeman and Company, New York.

Clark, A. J. (1972). An informal survey of multi-echelon inventory theory. *Naval Research Logistics Quarterly*, 19(4):621–650.

Clark, A. J. and Scarf, H. (1960). Optimal policies for a multi-echelon inventory problem. *Management Science*, 6(4):475–490.

Clausen, J., Larsen, A., Larsen, J., and Rezanova, N. J. (2010). Disruption management in the airline industry - concepts, models and methods. *Computers & Operations Research*, 37(5):809 – 821.

Cordeau, J.-F., Soumis, F., and Desrosiers, J. (2000). A benders decomposition approach for the locomotive and car assignment problem. *Transportation Science*, 34(2):133–149.

Edwards, J. S. (1983). A survey of manpower planning models and their application. *Journal of the Operational Research Society*, 34(11):1031–1040.

Eiselt, H. and Marianov, V. (2008). Employee positioning and workload allocation. *Computers & Operations Research*, 35(2):513–524.

Emmons, H. and Burns, R. N. (1991). Off-day scheduling with hierarchical worker categories. *Operations Research*, 39(3):484–495.

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(1):3–27. Timetabling and Rostering.

Flughafen München GmbH (2017). Statistischer jahresbericht 2017.

Fowler, J. W., Wirojanagud, P., and Gel, E. S. (2008). Heuristics for workforce planning with worker differences. *European Journal of Operational Research*, 190(3):724 – 740.

Fragnière, E., Gondzio, J., and Yang, X. (2010). Operations risk management by optimally planning the qualified workforce capacity. *European Journal of Operational Research*, 202(2):518 – 527.

Frey, M., Kolisch, R., and Artigues, C. (2017). Column generation for outbound baggage handling at airports. *Transportation Science*, 51(4):1226–1241.

Fündeling, C.-U. and Trautmann, N. (2010). A priority-rule method for project scheduling with work-content constraints. *European Journal of Operational Research*, 203(3):568 – 574.

Gans, N. and Zhou, Y.-P. (2002). Managing learning and turnover in employee staffing. *Operations Research*, 50(6):991–1006.

Gurobi Optimization, L. (2018). Gurobi optimizer reference manual.

Harvey, W. D. and Ginsberg, M. L. (1995). Limited discrepancy search. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, IJCAI'95, pages 607–613, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Jaillet, P. and Wagner, M. R. (2014). *Online Optimization—An Introduction*, chapter Chapter 6, pages 142–152.

Jiang, H. and Barnhart, C. (2013). Robust airline schedule design in a dynamic scheduling environment. *Computers & Operations Research*, 40(3):831 – 840.

Kiermaier, F., Frey, M., and Bard, J. F. (2016). Flexible cyclic rostering in the service industry. *IIE Transactions*, 48(12):1139–1155.

Koutsopoulos, H. N. and Wilson, N. H. (1987). Operator workforce planning in the transit industry. *Transportation Research Part A: General*, 21(2):127–138.

Kreipl, S. and Pinedo, M. (2004). Planning and scheduling in supply chains: An overview of issues in practice. *Production and Operations Management*, 13(1):77–92.

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

Martel, A. and Price, W. (1981). Stochastic programming applied to human resource planning. *Journal of the Operational Research Society*, 32(3):187–196.

Millar, H. H. and Kiragu, M. (1998). Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming. *European Journal of Operational Research*, 104(3):582–592.

Petersen, J., Sölveling, G., Clarke, J.-P., Johnson, D., and Shebalov, S. (2012). An optimization approach to airline integrated recovery. *Transporportation Science*, 46:482 – 500.

Pinker, E. J. and Shumsky, R. A. (2000). The efficiency-quality trade-off of cross-trained workers. *Manufacturing & Service Operations Management*, 2(1):32–48.

Powell, W. B. (2007). *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley Series in Probability and Statistics. Wiley-Interscience, New Jersey.

Price, W., Martel, A., and Lewis, K. (1980). A review of mathematical models in human resource planning. *Omega*, 8(6):639–645.

Purkiss, C. (1981). Corporate manpower planning: a review of models. *European Journal of Operational Research*, 8(4):315 – 323.

Puterman, M. (2014). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley-Interscience, New Jersey.

Reyck, B. D. and Herroelen, W. (1999). The multi-mode resource-constrained project scheduling problem with generalized precedence relations. *European Journal of Operational Research*, 119(2):538 – 556.

Ruf, C., Bard, J. F., and Kolisch, R. (2019a). Workforce capacity planning with hierarchical skills, long-term training, and random resignations.

Ruf, C., Schiffels, S., Kolisch, R., and Frey, M. (2019b). A data-driven approach for baggage handling operations at airports.

Ruxton, G. D. (2006). The unequal variance t-test is an underused alternative to student's t-test and the mann-whitney u test. *Behavioral Ecology*, 17(4):688–690.

Sandborn, P. A. and Prabhakar, V. J. (2015). The forecasting and impact of the loss of critical human skills necessary for supporting legacy systems. *IEEE Transactions on Engineering Management*, 62(3):361–371.

Sarna, S. K. (2013). Irregularities in blast furnace during operation. last accessed 2018-05-01.

Scarf, H. (1960). The optimality of (S, s) policies in the dynamic inventory problem. In Arrow, K., Karlin, S., and Suppes, P., editors, *Mathematical Methods in the Social Sciences*. Stanford University, Stanford.

Silver, E. A., Pyk, D. F., and Thomas, D. J. (2016). *Inventory and Production Management in Supply Chains*. CRC Press.

SITA (2018). Baggage report 2018.

Song, H. and Huang, H.-C. (2008). A successive convex approximation method for multistage workforce capacity planning problem with turnover. *European Journal of Operational Research*, 188(1):29–48.

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

Thyssenkrupp AG (2014). Erste Ofenreise nach 21 Jahren beendet: Größter Hochofen Europas wird modernisiert (first furnace campaign ended after 21 years: Europe's largest blast furnace is being modernized). last accessed 2018-06-06.

Valeva, S., Hewitt, M., Thomas, B. W., and Brown, K. G. (2017). Balancing flexibility and inventory in workforce planning with learning. *International Journal of Production Economics*, 183:194 – 207.

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.

Vanderbeck, F. (2000). On dantzig-wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. *Operations Research*, 48(1):111–128.

Vanderbeck, F. (2005). *Implementing Mixed Integer Column Generation*, pages 331–358. Springer US, Boston, MA.

Wang, J. (2005). *A review of operations research applications in workforce planning and potential modelling of military training*. DSTO Systems Sciences Laboratory Edinburgh, South Australia.

Welford, B. P. (1962). Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420.

Wirojanagud, P., Gel, E., Fowler, J., and Cardy, R. (2007). Modelling inherent worker differences for workforce planning. *International Journal of Production Research - INT J PROD RES*, 45:525–553.

Yan, S. and Tang, C.-H. (2007). A heuristic approach for airport gate assignments for stochastic flight delays. *European Journal of Operational Research*, 180(2):547 – 567.

Ye, L., Miao, S., and Yi, Y. (2018). Approximation approaches for inventory systems with general production/ordering cost structures. *Production and Operations Management*, 27(3):417–432.

Zülch, G., Rottinger, S., and Vollstedt, T. (2004). A simulation approach for planning and re-assigning of personnel in manufacturing. *International Journal of Production Economics*, 90(2):265–277.