Aalto University
School of Science

Master’s Programme in Mathematics and Operations Research

Computational models for evaluating contract structures under uncertainty

Alvar Wilhelmsson

Master’s Thesis
2025

Aalto University
School of Science

Author Alvar Wilhelmsson

Title Computational models for evaluating contract structures under uncertainty

Degree programme Mathematics and Operations Research

Major Systems and Operations Research

Supervisor Prof. Ahti Salo

Advisors D.Sc.(Tech.) Antti Punkka, Teemu Kinnunen M.Sc.(Tech.)

Collaborative partner SOK (Suomen Osuuskauppojen Keskuskunta)

Date November 21, 2025 **Number of pages** 70+14 **Language** English

Abstract

The retail sector utilizes a lot of part-time employees due to uncertainty in employee availability and variability in work demand. As many different contract structure configurations can have the same total work capacity, evaluating contract structures requires balancing cost and flexibility. Resourcing decisions should also consider operational needs, labor regulations, and the fixed cost of employees, costs independent of the number of hours worked.

Existing research on workforce planning in the retail sector predominantly focuses on short-term scheduling for fixed contract structures, with limited attention to long-term design and evaluation of alternative contract structures. Few approaches adequately capture the variable nature of work demand and the stochastic nature of staff availability, while adhering to the constraints of Finnish labor regulations.

This thesis develops computational models for evaluating contract structures under uncertainty. These models form a framework that consists of four phases: generation, planning, realization, and evaluation. Contract structures are generated using Dirichlet-distributed proportions for different contract types and filtered with constraints defined by the decision-maker. Contract structures are then scheduled algorithmically and tested in agent-based simulations that incorporate stochastic sickness and replacement processes. The resulting outputs are evaluated using cost and flexibility metrics, visualized using a Pareto front to highlight trade-offs between objectives.

The results indicate that cost-efficient contract structures rely primarily on mid-sized contracts of 15–20 hours per week. Moreover, contract structures with identical employee counts can differ substantially in flexibility, highlighting the importance of resource planning beyond simply adjusting staffing to demand. The results also show that neither an overreliance on large nor small contract types seems to be the best solution.

Keywords Stochastic agent-based simulation, workforce planning, resourcing, variable demand, cost and cost analysis, scheduling

Aalto-universitetet
Högskolan för
teknikvetenskaper

Författare Alvar Wilhelmsson	
Titel Beräkningsmodeller för utvärdering av kontraktstrukturer under osäkerhet	
Utbildningsprogram Mathematics and Operations Research	
Huvudämne Systems and Operations Research	
Övervakare Professor Ahti Salo	
Handledare Teknologie doktor Antti Punkka, Teknologie magister Teemu Kinnunen	
Samarbetspartner SOK (Centrallaget för Handelslagen i Finland)	
Datum 21 november 2025 **Sidantal** 70+14	**Språk** engelska

Sammandrag

Detaljhandelssektorn använder många deltidsanställda, på grund av osäkerheten i tillgänglighet av anställda samt variationen i arbetsmängd. Eftersom olika kontraktstrukturer kan ha samma totala arbetskapacitet, kräver utvärderingen av kontrakt strukturer att man balanserar flexibilitet och kostnader. Beslut om resursplanering borde också beakta operativa behov, arbetslagar, fackregler samt anställdas fasta kostnader, kostnader som inte beror på mängden arbetstimmar.

Existerande forskning i personalplanering fokuserar huvudsakligen på korttids schemaläggning, för en given kontraktstruktur. Långtidsplanering och utvärdering av kontraktstrukturer har fått relativt lite uppmärksamhet. Få modeller lyckas på ett tillfredställande sätt inkludera arbetsmängdens variation samt anställdas stokastiska beteende, samtidigt som de följer de begränsningar som finska arbetslagar och fackregler kräver.

I denna avhandling utvecklar vi beräkningsmodeller, som utvärderar kontraktstrukturer under osäkerhet. Modellerna skapar ett ramverk som består av fyra olika faser: generering, planering, utvärdering och utvärdering. Vi skapar kontraktstukturer på basis av Dirichlet-fördelade proportioner motsvarande kontrakttyper, som sedan filtreras enligt beslutsfattarens direktiv. Vi fördelar sedan arbetsturer till anställda i kontraktstrukturen med en algoritm. Det skapade schemat testar vi med en agent baserad simuleringsmodell. Simuleringsmodellen inkorporerar stokastiska sjukdomsprocesser samt ersättningsprocesser. Sjukdomsprocesserna beskriver hur anställda insjuknar och tillfrisknar, ersättningsprocesserna hur sjuka arbetares arbetsturer ersätts. Kontraktstrukturerna jämförs med två nyckeltal, flexibilitet och kostnad, vilket visualiseras med en Pareto front, som framhäver Pareto optimala kontraktstrukturer.

Resultaten visar att mellanstora kontrakt, mellan 15–20 timmar i veckan, är de mest lönsamma. Kontraktstrukturer med exakt lika många anställda kan skilja sig markant då det gäller flexibilitet, vilket framhäver värdet av en resursplanering som går utöver att enbart säkerhetsställa att mängden anställda är anpassad efter arbetsmängd. Därtill visar resultaten att varken för många små eller stora kontrakt verkar vara den bästa lösningen.

Nyckelord Stokastik agent baserad simulering, personalplanering, resursering, varierad efterfrågan, kostnad och kostnads analys, schemaläggning

Preface

I would like to thank my supervisor, Professor Ahti Salo, for his patience and insights, and my advisors, Doctor Antti Punkka and Mr. Teemu Kinnunen, for their valuable advice and guidance, without which the project would not have been possible.

I am grateful to SOK for providing both data and funding essential for the realization of this thesis, and my colleagues for their helpful discussions and advice.

Finally, I would like to thank my friends, family, and partner for being unbelievably patient and supportive during the writing process.

Helsinki, 21 November 2025

Alvar Wilhelmsson

Contents

Symbols and abbreviations

Symbols

E	Set of internal employees
E_{ext}	Set of external employees
E_r	Set of rental employees
D	Set of days
W	Set of weeks
T	Set of contract types
T_p	Shift state in the planning phase
T_r	Shift state in the realization phase
C''	Set of contract structures
c_i	Contract structure, $n_{i,t}$ is the number of employees of type t
cs	Current contract structure
$\mathrm{SIM}(c_i)$	Set of independent simulations for contract structure c_i
$P(c_i)$	Shift plan for contract structure c_i
p_s	Probability that an employee becomes sick
p_{cw}	Probability that an employee accepts a new shift
p_r	Percentage of shifts covered by rental employees
p_{ab}	Percentage of shifts covered by external employees
p_f	Percentage of shifts covered by internal employees
$r_{e,w}$	Deviation of employee e from contracted hours in week w

Abbreviations

KPI	Key Performance Indicator
LR	Labor Regulations (Finnish)
ABM	Agent-Based Modeling
DES	Discrete-Event Simulation
SOK	Suomen Osuuskauppojen Keskunta
MLE	Maximum Likelihood Estimator

1 Introduction

In the retail sector, the use of part-time employees is important. Because demand fluctuates, shift schedules often include shifts shorter than 7.5 hours in order to properly meet said demand. This makes it impossible to construct a schedule with only full-time employees without generating undertime. Undertime is defined as situations in which employees work fewer hours than specified in their contracts. Part-time employees provide flexibility, as Finnish labor regulations (LR) [16] allow them to work, for example, 35 hours in one week and 25 in the next, as long as they average their contracted hours. They are also an important resource when covering vacancies that arise due to sickness or other short-notice absences.

However, relying heavily on small part-time contracts is not practical. Such contracts are often unpopular among employees, and they are more expensive due to fixed costs such as work clothes and insurance. Thus, resource planning requires balancing several key performance indicators (KPIs). Contract structures should meet demand without undertime, while also remaining compliant with labor regulations, which constrain scheduling.

Prior research on workforce planning in the retail sector has focused primarily on short-term scheduling [17][14]. Studies on resourcing exist across multiple sectors [1][19][6], but we found no approaches that treat employees as individual entities or model differences in their behavior while studying contract structures. Furthermore, existing literature rarely considers how contract structures perform under the stochastic conditions in the retail sector. This thesis aims to address this research gap.

To do so, we propose a framework divided into four phases.

1. Contract structure generation, where alternative contract structures are generated and filtered through decision-maker-defined constraints.

2. Planning, where the feasibility of each structure is tested by scheduling all employees. This ensures the structure is viable and detects undertime.

3. Realization, where agent-based simulations are used to evaluate each structure's flexibility in responding to vacancies due to sickness.

4. Evaluation, where results from the previous phases are summarized into measures of cost, flexibility, and viability.

Together, these phases form a comprehensive approach for evaluating contract structures under uncertainty.

The remainder of this thesis is structured as follows. Section 2 reviews existing research on different methods used for workforce planning and motivates the chosen methodology. Section 3 introduces the models, parameters, and constraints used in the thesis. Section 4 provides computational results from an example case and validates the models' behavior. Section 5 examines different scenarios relevant to decision-makers. Section 6 concludes with a summary of findings, limitations, and directions for future work.

2 Literature review

2.1 Agent-based modeling in workforce planning

Agent-Based Modeling (ABM) is a simulation approach that models systems as collections of autonomous, decision-making entities called agents [2]. In our workforce planning problems, these agents can, for example, represent employees, managers, or tasks, each with their own behaviors, schedules, and constraints.

ABM is well-suited for addressing staffing problems because it models the heterogeneous behavior of employees well. Unlike traditional top-down models, ABM allows for individual behavior, such as employee availability, shift preferences, or illness, to influence the overall system [2].

This bottom-up perspective makes ABM a good tool for evaluating contract structures under uncertainty, balancing trade-offs between cost and flexibility. Furthermore, it integrates constraints such as union agreements well, which prevent workers from taking extra shifts even if willing, depending on the number of hours they have worked during the week, how many days they have worked in a row, and the total number of hours they have worked during the planning period.

In the retail sector, ABM has been explored by Siebers et al. They study the impact of people management practices on customer satisfaction [22], as well as the impact of human resource management practices on productivity [21]. They highlight that one of ABM's advantages is its intuitive design, as results arise from the interactions of agents, not macro-level dynamics. This makes ABM particularly well-suited for simulating interactions between workers and their influence on other workers' decision-making, with the significant limitation that it is difficult to empirically validate such models.

Other researchers have used ABM to study resourcing. Griffiths et al. [7] study the baseline number of nurses needed in a hospital, with flexible staff deployment, that is, floating staff between units and temporary hires. The purpose of their study is to identify the level of baseline staffing needed to meet fluctuating demand, while considering possible absences. They define each hospital unit as an agent, with stochastic demand for each shift. The goal is to meet the demand using the personnel scheduled for the shift, personnel from other shifts if available, and temporary hires. The costs are determined by overstaffing, additional costs from longer stays due to understaffing, and the additional costs associated with temporary hires. Griffiths et al. identify several limitations in their study, noting that the results are specific to the context of the three studied hospitals, exclude some potential costs from low baseline staffing, which could make higher baselines appear even more cost-effective, that the model does not fully account for efficiency or quality differences among types of temporary staff, and does not model long-term demand shifts. However, patterns were consistent across three hospitals, and secondary analyses, which included the adverse effects of using a lot of temporary staff, further supported higher baseline staffing. The study shows the feasibility of studying resourcing with ABM.

Kant et al. [10] study the workforce and labor dynamics at a macroeconomic scale. The model utilizes two types of agents, firms and individuals. Individuals have attributes such as age, gender, skills, salary, marital status, and number of dependents, some determined at their creation and others evolving. The attributes influence their decisions and career progression. Firms employ at least one managing director agent who makes decisions in response to changing market conditions. The study investigates how policy changes, such as modifications to labor laws, affect employment rates, employee utility, and firm performance over time. While the scope of the study is the labor market as a whole rather than a single business, it shows the feasibility of using ABM to model employee and employer behavior.

2.2 Alternate approaches to agent-based modeling

2.2.1 Stochastic programming

Pantuso [6] applies stochastic programming to the problem of a sports team's composition, studying the case of constructing a football roster. The model determines the optimal set of players to hire and retain, given uncertainty in their future performance and cost. Each player is characterized by attributes such as position, market value, and skill level, and the stochastic element is modeled using scenario-based variations in performance over time. The objective function is to maximize the team's quality over time, subject to constraints on roster size, positional requirements, and budget limits. While the application domain is professional sports, the model shows that stochastic programming can model trade-offs in cost and performance, handling both uncertainty and heterogeneous resources, in this case, the players. This can be considered workforce planning in the realm of professional sports, making the modeling techniques relevant to this thesis. However, the approach does not address unplanned absences, shift coverage, or legal labor constraints, which are relevant to this thesis.

Abernathy et al. [1] present a three-stage stochastic programming approach to nurse staffing and scheduling in hospitals. In the policy stage, general operational rules are established. The staff planning stage determines medium-term adjustments, including hiring, training, and nurse relocation, to meet the projected demand under uncertainty. Finally, the short-term scheduling stage schedules nurses using the outputs of the previous stages. Uncertainty in staffing needs is modeled via probabilistic demand forecasts. This method integrates long-term workforce planning with short-term scheduling within a single optimization framework. However, while the framework manages variability in demand well, it does not account for differences in nurse behavior, limiting its applicability to this thesis.

Parisio and Jones [17] present a stochastic optimization model for scheduling employees in the retail sector. It generates weekly rosters that schedule employees to meet demand, utilizing forecasted demand. The demand forecast is used to determine staffing requirements for time slots. The model is a mixed integer nonlinear problem that is solved using stochastic programming, which creates schedules that balance coverage and labor costs, constrained by contractual obligations. By modeling demand uncertainty, the approach avoids over- or understaffing. The authors demonstrate its effectiveness in a case study, showing that the model creates schedules that are both cost-efficient and able to handle fluctuations in customer demand. While the model is well-suited for scheduling problems where demand is forecasted probabilistically, it assumes a fixed contract structure, limiting its usefulness for this thesis. Furthermore, it does not consider sickness or replacement processes, central to this thesis.

Punnakitikashem et al. [19] present a stochastic programming model that models nurse scheduling and rescheduling under uncertain demand. The framework considers both long-term scheduling decisions, planned 4-6 weeks before the shift, and short-term scheduling decisions, adjustments made on the day of the shift, combining them into a single optimization framework. Uncertainty is modeled through patient demand scenarios, influencing required staffing levels in units. Sick leaves, days off, or other absences are also accounted for. The objective function minimizes the excess workload, subject to a budget constraint. Their results show improvements in balancing nurse workload and service quality. However, while the model schedules well under uncertainty in demand, it assumes a fixed contract structure and does not consider flexible contract structures, which limits its relevance to this thesis.

2.2.2 Discrete-event simulation

Forbus and Berleant present a discrete-event simulation (DES) model [4] that studies the allocation of physicians across clinic and surgical services in a pediatric hospital. The model simulates the full patient journey, from clinic visit through follow-up surgery, using historical data. The core simulation variable is physician time allocation, how many half-day shifts are devoted to clinic work versus surgery, and how this affects patient wait times across the system. They state that this design requires stakeholder input, as the current system allocates physicians for a full day of surgical or clinical work. The model is optimized to decrease wait times, with the interesting result of increasing clinic wait time while significantly decreasing surgery wait time. This DES framework demonstrates that scheduling decisions directly impact patient wait times. Its strength lies in maximizing the utilization of available resources, but it does not consider heterogeneous employee behavior, contract structures, or absences, key areas of our thesis.

Lacinova et al. (2022) introduce ENTIMOS [11], a discrete-event simulation model designed to support operational planning in multiple sclerosis infusion centers. The model was built in collaboration with clinical experts and parameterized using data from Charing Cross Hospital. Inputs include treatment-specific parameters, resources, nursing staff, and costs, while outputs include patient throughput, waiting times, queue sizes, resource utilization, and costs. The simulation revealed that the current capacity will not be able to meet projected demand. With no changes, average waiting times would exceed clinical recommendations by 8 days in the first year, with delays going up to 30 days after 30 months, threatening patient safety and timely treatment delivery. The model suggests that interventions, such as adding one infusion chair annually or redirecting a portion of patients, can avert these clinically significant delays. While ENTIMOS offers scenario analysis and operational insights, it is highly specialized to infusion clinic workflows and does not model staffing behaviors or contract flexibility. The DES framework models flows and resources well, but does not incorporate employee decision-making or heterogeneity, which our thesis aims to model.

2.3 Assessment of earlier research

We have shown that different approaches towards resourcing and scheduling have been explored, including DES, ABM, and stochastic programming. The usefulness of mathematical models in resourcing has been extensively explored in healthcare, with approaches using ABM [7], stochastic programming [1][19], and discrete event simulation [4][11]. Similar challenges are faced by the retail and healthcare sectors, such as variable demand, irregular working hours, and varying skill sets between workers. Differences mainly arise from contract structures, where the retail sector relies on part-time employees, and the healthcare sector does not. Thus, if such differences are considered, methodologies developed for the healthcare sector can be considered applicable to our thesis.

In the retail sector, workforce planning has been explored, utilizing ABM [21][22] and stochastic programming [17]. However, none of the studies we found focus on resourcing, showing a possible gap in the current research. While it is likely that private organizations have conducted internal research on workforce planning, such studies are typically not publicly accessible.

Giovanni's model [6] shows that modeling resourcing with heterogeneous employees is feasible using stochastic programming. The differences in manpower planning between the sports and retail sectors imply that adapting the methodologies developed is not feasible. Similar issues arise when adapting the methodology from WorkSim by Kant et al. [10], due to the differences between studying the labor market as a whole and a single business.

We find that while man-power planning has been explored using both DES and stochastic programming, the realities of the retail sector make them unsuitable tools for this problem. The issue largely lies in the heterogeneous nature of the employees and the fact that employee decisions are constrained both by the decisions of other employees and the labor union agreements. DES is not well-suited for representing employees as individual agents, but instead prefers to treat them as a resource pool that can be allocated to tasks.

Stochastic programming can consider employees as heterogeneous individual agents, but it faces two major limitations. First, the problem space quickly grows infeasibly large, as we must consider at a minimum each unique shift type per day, as well as the number of each shift type, along with the possible allocations of employees to those shifts. It is also difficult to model the time-dependency of sickness processes and replacement processes, due to their dynamic nature.

Griffith et al. [7] show that modeling resourcing needs of a single business is feasible using ABM. Furthermore, they show that it is well equipped at modeling fluctuating demand, possible absences, and countermeasures to absences, while considering costs. As our goal in this thesis is to minimize costs while maximizing the flexibility of a business's contract structure, this shows that ABM is well-equipped to model such problems. However, our thesis differs from it, as we consider individuals as agents. We cannot consider weekly or even daily demand as a set amount of hours, but must consider it a set of shifts of varying length. Furthermore, we should ensure the structure can handle the limitations imposed by Finnish labor regulations. Finally, we hypothesize that workers with different contract types behave differently and are thus not equally likely to accept additional shifts. Kant et al. [10] show that modeling individual workers as agents is feasible, further supporting our choice of utilizing ABM.

Taken together, these findings show that while stochastic programming and DES offer valuable insights into workforce planning, they fail to adequately capture the heterogeneity of employees and the variability in contract types in the retail sector. ABM, by contrast, allows us to model these dynamics at the employee level, making it the most appropriate method for this thesis.

3 Methodology

3.1 Problem description, approach and scope

Our goal is to study the viability of different contract structures in a retail business. We utilize the key metrics of cost and flexibility. Flexibility refers to the ability of a structure to accommodate unforeseen absences, which we limit to sickness-related cases. Cost includes fixed employee costs and penalties from under-scheduling relative to the employee's contract hours.

To accomplish the stated goals for a given planning period, we consider four different phases:

1. **Contract structure generation**, in which theoretical contract structures are created for comparative analysis.

2. **Planning**, where employee schedules are created in accordance with labor regulations. This occurs before the planning period.

3. **Realization**, where the created schedules are simulated to observe how they respond to unforeseen absences. This occurs during the planning period.

4. **Evaluation**, where the viability of each contract structure is assessed after the planning period.

We utilize decision-maker input, shift and sickness data, and parameters estimated based on the data. The process is summarized in Figure 1.

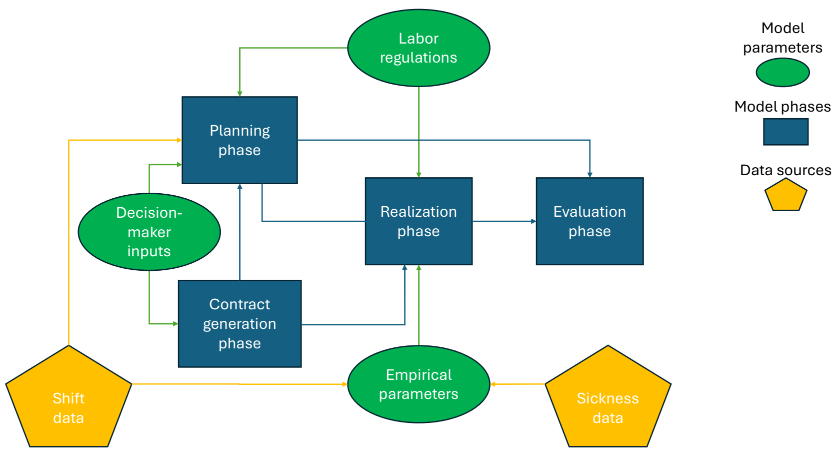

Figure 1: Overview of modeling approach.

The complexity of the processes described above necessitates that we limit the scope in the following ways:

1. **Demand estimation** is based exclusively on the previous year's realized data, rather than on predictive forecasting.

2. **Employee skill-sets and skill-based demand** are excluded from the model, given the assumption that employees can be trained as needed.

3. **Shift timing** is not differentiated when considering absences. For example, morning shifts may be harder to cover than evening shifts, but are treated equivalently. Furthermore, employee preferences are not included.

4. **Summer season** is excluded from the scope due to the confounding effects of summer holidays and elevated demand. The primary resourcing question for the summer season is how many seasonal workers to hire, unrelated to the number of permanent employees in the contract structure.

3.2 Data collection and processing

We utilize data provided by our partner SOK (Suomen Osuuskauppojen Keskunta), the client of this thesis. They have provided real data on work shifts, both at the planning phase, around three weeks before the realization of the shift, and at the realization phase, describing what actually happened. Finally, they provide data on the length of sick leaves.

The shift data contains information on what task the employee was performing, including different work-related tasks, and absences such as sick leaves, vacation days, and unpaid days off. The data only contains information regarding shifts performed by the employee. A date without shifts for a certain employee denotes a day off. The shift data is our basis for work demand, giving us an accurate estimate of both the number of shifts and their duration. Furthermore, the shifts are utilized when calculating parameters.

It should be noted that the dataset contains information not only on the business site to be studied, but also on other branches within the region. This broader dataset allows the chain-wide estimation of certain parameters, as further clarified in Section 3.4. An overview of the data can be found in Table 1.

Table 1: Overview of data provided by SOK.

Data source	**Contents**	**Usage in model**
Planned shifts	Shifts scheduled about three weeks before realization	Demand modeling and parameter estimation
Realized shifts	Final realized schedule	Parameter estimation
Sick leave data	Absence records, including length of sick leave	Estimation of length of sick leaves
Shift metadata	Task type (work duty, sick leave, vacation) and days off	Parameter estimation

3.3 Finnish labor regulations

The regulations that must be followed are determined by Palvelualojen ammattiliitto (PAM) [16], the labor union for retail employees. They define several rules regarding scheduling that heavily impact our model, as the company is subject to penalties upon breaking them. Another important factor is overtime rules, which guide company policy. Henceforth, they will be referred to using the common shorthand LR.

The main constraints apply during the planning stage. An employee can be scheduled for at most 48 hours per week without incurring overtime pay. They may average no more than five shifts per week and must be scheduled for, on average, at least their contracted weekly hours. Going below in one week is allowed if it is compensated for in another week. If demand is insufficient to meet these hours, the employee is nevertheless compensated as if they had been scheduled per their contract. Employees have a minimum number of weekends off per year, determined by the labor regulations. However, we do not account for these constraints in the thesis, as the quota is relatively low, vacation periods count toward the entitlement, and the quota is for the whole calendar year, introducing unknowns such as previous and future weekends off. The labor regulations also determine how many vacation days employees have.

3.4 Parameter estimation

Using the data described in Section 3.2, we estimate the parameters for the simulation model, including the probability that an employee falls ill, the distribution of the length of sick leave, the probability that an employee accepts a vacant shift, and the probability that an employee of another business within the company accepts a vacant shift. The variables used in calculating the parameters are defined in Table 2.

Table 2: Variables used for parameter estimation

Symbol	**Definition**
$E = \{e_1, \ldots, e_m\}$	Set of employees in the business
$E_{ext} = \{e_{m+1}, \ldots, e_{m+n}\}$	Set of external employees recorded in the data, a subset of all theoretically available external employees
$E_r = \{e_{m+n+1}, \ldots, e_{m+n+k}\}$	Set of rental workers recorded in the data, a subset of all theoretically available rental employees
$D = \{d_1, \ldots d_n\}$	Set of days (time frame of the analysis)
$T_p(e, d) \in \{\text{at work, sick, on vacation}, \emptyset\}$	State of shift (planning stage)
$T_r(e, d) \in \{\text{at work, sick, on vacation}, \emptyset\}$	State of shift (realization stage)
$l(e, d) \in \mathbb{Z}^+$	The length of a recorded sick leave for employee e starting on day d

3.4.1 Distribution of sick leave durations

We estimate the empirical probability distribution of sick leave durations as relative frequencies. To ensure that the absence begins on the first day recorded, we only include cases where the employee was at work on the day before the sick leave. Due to this limitation, we use data from all branches for this parameter. Furthermore, we exclude cases exceeding seven days. Formally, the relative frequency of a sick leave of length k is defined in Equation (1).

$$\hat{p}(k) = \frac{|\{(e,d) \mid e \in E, d \in D, l(e,d) = k, T_r(e,d-1) = \text{at work}\}|}{|\{(e,d) \mid e \in E, d \in D, 1 \le l(e,d) \le 7, T_r(e,d-1) = \text{at work}\}|}, \quad k = 1,\dots,7. \tag{1}$$

Equation (1) yields the relative frequency of a sick leave of length k, up to 7 days, while excluding longer absences, which are handled instead by long-term solutions such as temporary employees. The resulting discrete distribution of the durations of sick leaves is defined by Equation (2), i.e.,

$$S = \cup_{k=1}^{7} \{\hat{p}(k)\}. \tag{2}$$

3.4.2 Probability an employee falls sick

To estimate the probability that an employee falls sick, p_s, we first calculate the percentage of shifts lost due to sickness. Specifically, we define the sickness absence rate as the percentage of planned shifts that were subsequently reported as sick leaves on the realized schedule. This provides an estimate of the probability that an employee scheduled to work on a given day is unable to do so due to sickness, defined in Equation (3). Due to possible variation between branches and an adequate amount of data from the business, we only use data from the business to estimate the parameter.

$$p_{\text{sick}} = \frac{\left|\{e,d \mid e \in E, d \in D, T_p(e,d) = \text{at work} \wedge T_r(e,d) = \text{sick}\}\right|}{\left|\{e,d \mid e \in E, d \in D, T_p(e,d) = \text{at work}\}\right|}. \tag{3}$$

To calculate the probability of an employee falling sick, p_s, we make the following assumptions: that employees are equally likely to fall sick on any given day and that employees fall sick at the same rate. Given the assumptions and the previously defined p_{sick} and S, the probability of an employee falling sick can be calculated as defined in Equation (5).

$$p_{sick} = p_s \cdot \mathbb{E}[S], \tag{4}$$

$$p_s = \frac{p_{sick}}{\mathbb{E}[S]}. \tag{5}$$

3.4.3 The probability that an employee accepts a new shift

When estimating the probability that an employee accepts a vacant shift, we account for the following factors:

1. **Availability**: The ability to take the shift due to not being sick, on vacation, or on an existing work assignment.

2. **Shift status**: Whether the shift has already been filled by another internal employee, external employee, or rental worker.

3. **Weekday**: The day of the week, as willingness to accept additional shifts may vary across weekdays.

4. **Contract constraints**: The employee's contract hours, which may affect the acceptance rate.

To ensure we properly account for all factors, we estimate the probability using maximum likelihood estimation (MLE) [3]. The maximum likelihood estimator (MLE) of a parameter θ is defined as

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta \in \Theta} L(\theta \mid x_1, \ldots, x_n),$$

where $L(\theta \mid x_1, \ldots, x_n) = \prod_{i=1}^{n} f(x_i \mid \theta)$ is the likelihood function [3].

We define p as the probability of an employee accepting a shift, our unknown, n_d as all offers given on a given day d, ea_d as all employees available to cover a shift on a given day d, es_d as all employees that agreed to cover a shift on a given day d, and r_d as the number of shifts covered by external employees or rental workers.

As we assume an offer is made for every case of sickness, n_d can simply be calculated

$$n_d = |\{e | e \in E, T_p(e, d) = \text{at work} \wedge T_r(e, d) = \text{sick}\}|.$$

The number of employees available is the set of internal employees, not on vacation, at work, or sick, defined as:

$$ea_d = |\{e | e \in E, T_r(e, d) = \emptyset \wedge T_p(e, d) = \emptyset\}|.$$

Per interviews with store managers, we know that employees working in the realization phase, without being scheduled for a shift, indicate replacements for absences. Thus, we can define the number of replacements on a given day as:

$$es_d = |\{e | e \in E, T_r(e, d) = \text{at work} \wedge T_p(e, d) = \emptyset\}|.$$

Lastly, the number of shifts filled by external workers reduces to

$$r_d = |\{e | e \in E_{ext} \cup E_r, T_r(e, d) = \text{at work} \wedge T_p(e, d) = \emptyset\}|.$$

If all vacancies were filled, $n_d = r_d + es_d$, including cases where external workers filled some of the shifts, the exact number of internal employees who would have accepted is unknown. Thus, we know it is at least ea_d, but not the exact count.

Therefore, this scenario can be modeled as a right-censored binomial [3], where the likelihood contribution for day d is the probability that at least ea_d employees accept a new shift. This can be modeled using the inverse cumulative probability of the binomial distribution [3], which yields the likelihood function presented in Equation (6). We denote such days $D_{unknown}$, i.e.,

$$f(es_d, ea_d, |p) = 1 - \sum_{i=0}^{es_d - 1} [\binom{ea_d}{i} * p^i * (1-p)^{ea_d - i}] \mid d \in D_{unknown}. \tag{6}$$

If vacancies remain unfilled, $n_d > r_d + es_d$, we do not need to make assumptions, as the number of employees willing to accept a shift is known. Thus, the likelihood can be modeled as a binomial [3], with the likelihood defined in Equation (7). We denote such days D_{known}, i.e.,

$$f(es_d, ea_d | p) = \binom{ea_d}{es_d} * p^{es_d} * (1-p)^{ea_d - es_d} \mid d \in D_{known}. \tag{7}$$

Thus, given the sets of days $D_{unknown}$ and D_{known}, and the respective relevant likelihood functions defined in Equations (6) and (7), we can calculate the full likelihood for the set $D = D_{par} \cup D_{obs}$ as the product, defined in Equation (8). Days where $n_d = 0$ or $ea_d = 0$ do not need to be included, as no employees could make decisions on such days. Thus,

$$\begin{aligned} L(p | es_d, ea_d) = & \prod_{d \in D_{known}} [\binom{ea_d}{es_d} * p^{es_d} * (1-p)^{ea_d - es_d}] \times \\ & \prod_{d \in D_{unknown}} \{1 - \sum_{i=0}^{es_d - 1} [\binom{ea_d}{i} * p^i * (1-p)^{ea_d - i}]\}. \end{aligned} \tag{8}$$

The full likelihood, defined in Equation (8), can be maximized to estimate p. There is no simple numerical solution, but an optimizer can maximize p, providing a solution, per the form defined in Equation (9),

$$p = \arg\max_p L(p | es_d, ea_d). \tag{9}$$

To account for timing, we partition the weekdays into three sets:

$$W_1 = \{\text{Mon, Tue, Wed, Thu}\}, \quad W_2 = \{\text{Fri, Sat}\}, \quad W_3 = \{\text{Sun}\}.$$

We assume that W_1 behaves like regular weekdays, W_2 (weekend shifts) are harder to cover, and W_3 (Sundays) are treated separately due to higher pay in Finland.

Similarly, to account for contract constraints, we partition the employees into five sets according to their weekly contract hours, with 37.5 hours per week being the maximum contract size allowed by labor regulations:

$$C_1 = [0, 10], \quad C_2 = (10, 20), \quad C_3 = [20, 30), \quad C_4 = [30, 37.5), \quad C_5 = [37.5].$$

We hypothesize that employees in C_1 are less likely to accept extra shifts due to external obligations, such as studies. Those in C_4 are also less likely due to the smaller relative financial incentive from additional shifts.

Combining the two groups, we estimate 15 probabilities, one for each set, which is formalized in Equation (10), here we note that employees not belonging to the studied contract type group C_i are considered external employees. Given the number of groups, we elected to use data from all branches when estimating this parameter to ensure a sufficiently large data amount for all groups. Furthermore, it guarantees the anonymity of employees, as a branch could have only one employee belonging to a given group. However, given a set of branches and the size of the groups, no such cases occur.

$$p_{cw} = \arg\max_{p} L(p \mid ea_d \in C_i, es_d \in C_i, d \in W). \tag{10}$$

3.4.4 Percentage of vacant shifts covered by internal employees, external employees, rental workers, or not covered.

As no data is available on the availability of rental workers, our most accurate estimate is the percentage of shifts covered by a rental worker, defined in Equation (11). Given the variance in the availability of rental workers depending on location, only data from the business studied is considered, i.e.,

$$p_r = \frac{|\{e, d \mid e \in E_r, d \in D, T_r(e, d) = \text{at work}\}|}{\left|\{e, d \mid e \in E, d \in D, T_p(e, d) = \text{at work} \wedge T_r(e, d) = \text{sick}\}\right|}. \tag{11}$$

The data on what employees from other branches are available to cover shifts is limited. We only know the set of employees who have previously covered shifts in the studied business, not the set of employees in other branches willing to cover shifts. Thus, given the lack of information, our best estimate is the percentage of shifts covered, defined in Equation (12). Given the variance in the availability of such resources between branches, primarily due to location, only data from the business studied is considered, i.e.,

$$p_{ab} = \frac{|\{e, d \mid e \in E_{ext}, d \in D, T_r(e, d) = \text{at work}\}|}{\left|\{e, d \mid e \in E, d \in D, T_p(e, d) = \text{at work} \wedge T_r(e, d) = \text{sick}\}\right|}. \tag{12}$$

To utilize the two parameters described previously, we must also calculate the percentage of vacant shifts covered by internal employees. We define the percentage in Equation (13), only utilizing data from the studied business, to ensure the results are comparable to p_r and p_{ab}, i.e.,

$$p_f = \frac{\left|\{e, d \mid e \in E, d \in D, T_r(e, d) = \text{at work} \wedge T_p(e, d) = \emptyset\}\right|}{\left|\{e, d \mid e \in E, d \in D, T_p(e, d) = \text{at work} \wedge T_r(e, d) = \text{sick}\}\right|}. \tag{13}$$

Finally, we can define the percentage of shifts not covered, p_{nf}, as defined in Equation (14).

$$p_{nf} = 1 - p_f - p_{ab} - p_r \tag{14}$$

3.5 Model design overview

Given the described approach in Section 3.1, to evaluate the efficiency of a given contract structure, the model is divided into four stages: contract structure generation, planning, realization, and evaluation, with each phase described in detail below.

3.5.1 Contract structure generation phase

The goal of the contract structure generation stage is to generate a set of contract structures, which we compare to the current one. The set of generated contracts can be constrained according to the decision-maker. Constraints include the maximum and minimum number of employees, and the maximum and minimum number of contract types (full-time and various types of part-time contracts).

3.5.2 Planning phase

The goal of the planning phase is to determine whether all shifts can be scheduled in advance, without generating undertime. We utilize an algorithmic approach, utilizing a recursive algorithm, stopping at the first feasible solution, with corrections post-recursion. It sequentially assigns each available shift to a worker while respecting contractual and regulatory constraints.

Our goal is not to produce an optimal shift schedule, but to test the feasibility of the contract structure. If all shifts can be allocated without violating constraints, the structure is considered viable for the realization and evaluation phases.

Heuristic algorithms have been successfully applied to workforce scheduling problems. Montoya and Mejía [14] propose an approach that minimizes the total cost of worker shortages and excesses while maintaining a constant number of shifts per employee. Their results demonstrate that the method consistently produces optimal or near-optimal solutions, providing empirical support for the validity of using an algorithmic scheduling strategy in this thesis.

3.5.3 Realization phase

The realization stage models how the planned schedule unfolds under uncertainty in employee availability. We employ an agent-based simulation model with a discrete-time resolution of one day, consistent with the precision of the available data. ABM has been utilized previously for problems in the retail sector.[21][22].

Bonabeau [2] describes agent-based modeling as a simulation technique in which results emerge from the behavior of autonomous agents within the model, where the agents make independent decisions. It is considered more of a mindset, not a technique, where the system is described from a bottom-down perspective and even simple models can exhibit complex patterns [2]. This model utilizes employees as agents, each of whom may independently experience sickness or decide to accept additional shifts.

- Sickness dynamics: Agents can become sick with a probability p_s estimated from empirical data. A sick agent remains unavailable for a fixed number of days, with the duration drawn from S.

- Shift acceptance: If a shift becomes vacant, eligible agents decide independently with probability p_{cw} whether to accept the shift, subject to LR restrictions. Each open shift can be filled by at most one agent.

The simulation is executed with 1,000 Monte Carlo iterations, which we verify is sufficient to reduce random variance in Section 4.4.2.

3.5.4 Evaluation phase

The outcomes of the simulations are evaluated using two performance measures:

1. Cost – defined as the fixed price of the contract structure, in conjunction with possible excess costs from undertime, that is, employees not being scheduled enough hours to meet their contract.

2. Flexibility – defined as the percentage of vacant shifts that are successfully covered.

The trade-off between these two objectives is analyzed using a Pareto front. This lets decision-makers identify personnel structures that strike a balance between cost and flexibility, aligning with their priorities. For instance, larger stores may prioritize cost, whereas smaller ones may prioritize flexibility.

3.6 Contract structure generation phase

Our goal is to generate a set of contract structures, denoted as $C'' = \{c_1, \ldots, c_k\}$, which we compare to the current structure, denoted as cs. Each contract structure describes the number of employees per contract type.

3.6.1 Notation

- $T = \{t_1, \ldots, t_m\}, t_i \in [0, 37.5]$, defines each unique contract type, marked as their contract hours, where m is the number of contract types and M is the set of indexes $\{1, \ldots m\}$. Furthermore, we order the contract types from largest to smallest i.e., $t_i > t_J \Leftrightarrow i < j$.

- $c = (n_1, \ldots, n_m) \in \mathbb{Z}^+$, is a contact structure, defined as a vector where n_i is the number of employees of type t_i

- G, the desired sum of contract hours for the planning period, defined as a percentage of the average weekly demand, with the percentage determined by the decision maker.

- δ, the allowed deviation from the goal G, defined as a percentage.

- N_{min}, N_{max}, lower and upper bounds for employee count $\sum_{i=1}^{m} n_i$.
- $L_{min}(t_i), L_{max}(t_i)$, lower and upper bounds for employee contract type t_i.

3.6.2 Generation algorithm

We generate type proportions $d_i = (\pi_1, \dots, \pi_m)$, that are the generated proportions of contract types, using a Dirichlet distribution [5] defined in Equation (15). We assign equal values to the weight parameter α, as we do not want to bias towards certain contract structures.

$$d_i \sim \text{Dirichlet}(\alpha), \quad \alpha = (\alpha_1, \dots, a_m) = (1, \dots, 1) \ . \tag{15}$$

Let the set $P = \{d_1, \dots, d_l\}$ be the group of l generated type proportions. We exclude distributions that represent clearly unrealistic scenarios by filtering out all cases in which more than 60 percent of employees are assigned to a single contract type. In addition, we constrain the share of full-time contracts to a maximum of 33 percent. We set contract type n_1 to equal full-time employees, and thus limit the corresponding value d_1, yielding the final set

$$P' = \{d \in P \mid \max(d) \leq 0.6 \ \wedge \ (d_1) \leq 0.33\},$$

with $r \leq l$ entries.

Given the set of proportions $P' = \{d_1, \dots, d_r\}$, we convert them to a set of corresponding contract structures $C = \{c_1, \dots, c_r\}$ using the following procedure. Let $d_i = (\pi_1, \dots, \pi_m)$ be the target distribution, $c_i = (n_i)_{i \in M} = (0, \dots, 0)$ the contract vector, and $n = \sum_{i \in M} n_i$ the current number of assigned contracts. We define the function

$$S(n) = \arg\max_{i \in M} \big((n+1)\pi_i - n_i\big).$$

Increment case. If the total contract hours do not exceed the goal G, taking deviation δ into account,

$$\sum_{i \in M} n_i t_i \leq G + \delta,$$

choose the smallest index among the maximizers and increment that entry:

$$t^\star(n) = \min S(n), \qquad c_i \ \leftarrow \ c_i + e_{t^\star(n)}.$$

Decrement case If the total contract hours exceeds the goal G taking deviation δ into account,

$$\sum_{i \in M} n_i t_i > G + \delta,$$

choose the smallest contract type in use and decrement that entry:

$$t^\star = \min\{t_i | n_i > 0\}, i \in M, \qquad c_i \ \leftarrow \ c_i - e_{t^\star}.$$

To avoid potential infinite loops, once a decrement step has occurred, the procedure does not perform increments and continues with decrements only until either the end condition is met,

$$\sum_{i \in M} n_i t_i \in [G - \delta, G + \delta],$$

or the total contract hours fall below,

$$\sum_{i \in M} n_i t_i < G - \delta.$$

Initialization (first choice). When $n = 0$ and $c = 0$, we maximize p_t. Let

$$S(0) = \arg\max_{i \in M} p_i, \qquad t^\star(0) = \min S(0), \qquad c \leftarrow c + e_{t^\star(0)}.$$

End condition. The procedure ends once the total contracted hours are sufficiently close to the target level G:

$$\sum_{i \in M} n_i t_i \ \in \ [G - \delta, G + \delta].$$

If the decrement phase causes the total contracted hours to fall below $G - \delta$ without entering this interval, the algorithm also terminates to avoid potential infinite loops.

3.6.3 Final selection

Given the set of contracts C described in the Section 3.6.2, we filter out all contracts that do not fulfill the following two conditions:

1. **Total number of employees.** For every contract structure $c \in C$, the total number of employees must lie within

$$N_{\min} \ \leq \ \sum_{i \in M} n_i \ \leq \ N_{\max}.$$

2. **Per-contract-type bounds.** For every contract structure $c \in C$ and each contract type t_i, the number of employees assigned to type t_i must lie within

$$L_{\min}(t_i) \ \leq \ n_i \ \leq \ L_{\max}(t_i).$$

After applying these constraints, we obtain the filtered set of feasible contracts

$$C' \ = \ \left\{ c \in C \ \middle| \ N_{\min} \leq \sum_{i \in M} n_i \leq N_{\max}, \ L_{\min}(t_i) \leq n_i \leq L_{\max}(t_i) \ \ \forall i \in M \right\}.$$

Given the set $C' = \{c_1, \ldots, c_s\}$ with $s \leq r$ entries, we have a corresponding set of target distributions P''. Utilizing Jensen-Shannon divergence [12], we aim to find the k most different contract structures, selecting the ones that are most dissimilar to the rest. This process is defined in the algorithm below.

Let $P' = \{d_1, \ldots, d_s\}$ and define the Jensen-Shannon distance $d(x, y) = \sqrt{\mathrm{JS}(x\|y)}$. For each j set

$$\mathrm{MinD}(j) \;=\; \min_{\substack{\ell=1\\ \ell\neq j}}^{s} d\big(p^{(j)}, p^{(\ell)}\big).$$

We select the k targets that are most dissimilar by

$$S^\star \;=\; \text{indices of the top-}k \text{ values of } \mathrm{MinD}(j).$$

If ties occur, we choose the smallest index. Giving us the set:

$$C'' = C'[S^\star].$$

The set C'' thus represents the k most diverse contract structures, which are compared against the current structure cs.

3.7 Planning phase

The planning phase involves scheduling shifts for employees in advance, creating a feasible plan for the given planning horizon. In practice, companies typically schedule shifts approximately three weeks in advance. Our model simplifies this by assigning all shifts at once for the whole planning window. As our demand, the previous years' planned shifts T_p are used.

3.7.1 Vacation allotment

Winter vacation During the planning period, some employees take one full week of winter vacation. We assign them before allocating shifts. Let $V = \{v_1, \ldots, v_m\}$ be the set of weeks studied, and $v_e \in \{1, \ldots, W\}$ denote the set of assigned vacation weeks for employees $e \in E$, where W is the number of weeks in the planning period. Vacations are allocated according to the following rules:

1. Weeks with lower demand are prioritized, such that

   $$\Pr(v_e = w) \propto \frac{1}{d_w},$$

   where d_w denotes demand in week w.

2. Full-time employees cannot be assigned to the final week, i.e.,

   $$v_e \neq W \quad \text{if employee } i \text{ is full-time.}$$

3. The order of assignment is random.

Annual leave Under the LR agreement [16], employees earn a day of annual leave in accordance with a table defined in the LR agreement [16]. Days are earned after a certain number of hours worked, and should be scheduled as soon as possible. The number of hours needed for the next annual leave is determined by the total number of hours worked during the calendar year. Because our time frame is shorter, we adopt the following simplification: an employee earns one day of annual leave for every 200 working hours. Counted hours include all planned working hours and all paid vacation hours that accrue leave. Annual leave is distributed as part of the shift assignment process to mimic real-world decision-making.

3.7.2 Shift assignment

To schedule shifts for employees, we define several concepts. Given the set of employees E defined previously, for each e, let $h_e \in \mathbb{R}$ be the hour balance (positive for overtime, negative for undertime), $ch_e \in \mathbb{R}$ be the contract hours, and $a_e \geq 0$ be the hours accrued toward annual leave [16]. We initialize per employee as

$$a_e^{(0)} \sim \text{Unif}[0, 200) \text{ hours},$$

with $a_e^{(0)}$ as the hours already accumulated toward the annual leave.

We use the previously defined set of weeks in the planning horizon $V = \{v, \ldots, v_m\}$. For each $v \in V$, let D_v denote the set of calendar days in week v, with index $d \in D_v$. For each day $d \in D_v$, let T_d denote the set of shifts scheduled on day d, with index $t \in T_d$. We denote $x_{e,d,t} \in \{0, 1\}$ the shift assignment variable indicating whether employee e is assigned to shift t on day d. We denote each shift's duration $h_t, t \in T$. We define the variable $xp_{e,d}$ as the number of consecutive shifts the employee has worked before day d. Formally,

$$xp_{e,d} = \begin{cases} 0, & \text{if } \sum_{t \in T_d} x_{e,d,t} = 0 \\ xp_{e,d-1} + 1, & \text{otherwise} \end{cases}.$$

We distribute shifts using a recursive allocation algorithm, which distributes shifts one week at a time, to ease computational load. As an initial step, we incorporate winter vacations by enforcing the constraint

$$x_{e,d,t} = 0 \quad \forall e \in E,\ \forall d \in D_{v_e},\ \forall t \in T_d.$$

We further constrain the allocation problem to ensure labor regulations (LR) [16] are followed. For convenience, we define

$$LR(x, xp, E, D, T, V),$$

as the set of constraints that enforce LR compliance. This function encapsulates all relevant LR requirements. The requirements are maximum weekly shift limit, consecutive shift limit, daily shift limit, total hours limit, and weekly hours limit, which are defined below.

1. **Weekly shift limit:** For each employee e the the weekly number of shifts must satisfy

$$\sum_{d \in D_v} \sum_{t \in T_d} x_{e,d,t} \;\leq\; m_w = 5 \quad \forall e \in E,\ \forall v \in V,$$

which is a slight simplification of the LR requirement, being at most five shifts per week on average. The requirement is simplified to improve computational performance and quality of results.

2. **Consecutive shift limit:** For each employee e, day d, the number of consecutive shifts must satisfy

$$xp_{e,d} \leq m_r = 8, \forall e \in E, \forall d \in D_v, \forall v \in V.$$

3. **Total hours limit:** For each employee e, the cumulative hours worked over the planning horizon must satisfy

$$\sum_{v \in V} \sum_{d \in D_v} \sum_{t \in T_d} h_t x_{e,d,t} \;\leq\; h_{\max}, \quad \forall e \in E,$$

where $h_{max} = 37.5 * |V|$. This is set as the max to avoid paid overtime.

4. **Weekly hours limit:** For each week $v \in V$, the weekly hours assigned to employee e must satisfy

$$\sum_{d \in D_v} \sum_{t \in T_d} h_t x_{e,d,t} \;\leq\; h_{\text{week}} = 48, \quad \forall e \in E, \forall v \in V.$$

5. **Daily shift limit** For each day d, employee e, the number of shifts worked must satisfy

$$\sum_{t \in T_d} x_{e,d,t} = 1, \quad \forall e \in E, \forall d \in D_v, \forall v \in V.$$

We define the number of hours we desire the employee e to work $g_{e,v}$ in a given week v as the combination of their contract hours and hour balance h_e

$$g_{e,v} = ch_e + h_e.$$

Annual-leave trigger and allocation. Let $y_{e,v} \in \{0, 1\}$ be the indicator showing whether an employee has earned an annual leave, $al_{e,d} \in \{0, 1\}$ the assignment variable determining which day the employee is assigned annual leave.

Trigger. When $a_e \geq 200$, we attempt to schedule exactly one annual-leave day within the current week v:

$$a_e \geq 200 \;\Rightarrow\; \text{set } y_{e,v} = 1.$$

Weekly cap. At most a fifth of employees can be on annual leave in week v, added to improve performance and quality of results:

$$\sum_{e \in E} y_{e,v} \;\le\; \left\lfloor \frac{|E|}{5} \right\rfloor .$$

Linking constraint. A weekly leave gives exactly one day of annual leave:

$$y_{e,v} \in \{0,1\}, \quad al_{e,d} \in \{0,1\}, \qquad \sum_{d \in D_v} al_{e,d} \;=\; y_{e,v} \quad \forall e \in E.$$

Effects if leave is granted ($y_{e,v} = 1$). We adjust the weekly targets and constraints:

$$g'_{e,v} \;=\; g_{e,v} - 7.5, \qquad \sum_{d \in D_v} \sum_{t \in T_d} x_{e,d,t} \;\le\; m_{e,v} = 4.$$

If $y_{e,v} = 0$, the weekly shift constraints and target remain, as defined in Section 3.7.2.

Day selection for leave. After scheduling all shifts for the week, assign the annual leave day to a randomly selected non-working day:

$$\text{pick } d^\star \in D_v \text{ with } x_{e,d^\star,t} = 0 \; \forall t \in T_{d^\star}, \quad al_{e,d^\star} = 1.$$

Counters and hour balance. Upon confirming the leave day,

$$a_e \leftarrow a_e - 200, \qquad h_e \leftarrow h_e + 7.5,$$

and, per LR [16], the 7.5 hours contributes to hours worked h_e but not to future annual-leave a_e.

3.7.3 Shift allocation algorithm

We assign shifts one week at a time, one employee at a time, assigning shifts that are close to each employee's contractual goal hours, g_{ev}. For a given employee e, we perform a recursive search over all daily shifts. At each step, we either assign one feasible shift of duration h_t (or assign none) and then perform a recursive step. Each assignment is immediately checked for LR compliance. When a solution for the employee is found, we update the referenced state variables: the consecutive-shift counter xp and the shift assignment variable x. The process for one week is formalized in Algorithm 1, where δ is our allowed deviation from the goal.

Algorithm 1 Shift allocation algorithm

Require: $x, xp, E, V, D_v, T_d\, g, \delta, e_{\text{cur}}, v_{\text{cur}}$

$d_{\text{cur}} \leftarrow \min(D_{v_{\text{cur}}})$
procedure ALLOCATE$(x, xp, e_{\text{cur}}, v_{\text{cur}}, d_{\text{cur}})$
 ▷ Termination: hit target weekly hours (within tolerance)
  if $\left|\sum_{d\in D_{v_{\text{cur}}}} \sum_{t\in T_d} h_t x_{e_{\text{cur}},d,t} - g_{e_{\text{cur}},v_{\text{cur}}}\right| \le \delta$ **then**
    return $(\textbf{True}, x, xp)$
  end if
 ▷ Candidates for today: either no shift, or one LR-feasible shift
  $C \leftarrow \{\texttt{none}\} \cup \{t \in T_{d_{\text{cur}}} \mid x_{t,e,d_{cur}} = 0, \forall e \in E\}$
  for all $c \in C$ **do**
    if $c = \texttt{none}$ **then**
      $xp' \leftarrow \texttt{updateConsecutive}(xp, e_{\text{cur}}, d_{\text{cur}}, 0)$
      $(ok, x', xp') \leftarrow$ ALLOCATE$(x, xp', e_{\text{cur}}, v_{\text{cur}}, \text{next}(d_{\text{cur}}))$
      if ok **then**
        return $(\textbf{True}, x', xp'')$
      end if
    else
      $x' \leftarrow x; \quad x'_{e_{\text{cur}},d_{\text{cur}},c} \leftarrow 1$
      $xp' \leftarrow \texttt{updateConsecutive}(xp, e_{\text{cur}}, d_{\text{cur}}, 1)$
      if $\texttt{LR}(x', xp', E, D, T, V)$ **then**
        $(ok, x'', xp'') \leftarrow$ ALLOCATE$(x', xp', e_{\text{cur}}, v_{\text{cur}}, \text{next}(d_{\text{cur}}))$
        if ok **then**
          return $(\textbf{True}, x'', xp'')$
        end if
      end if
    end if
  end for
  return $(\textbf{False}, x, xp)$
end procedure

3.7.4 Empty-shift backfill

After Algorithm 1 assigns shifts for a week, some shifts may remain empty. To close these gaps, we apply a greedy backfill for the week. For every empty shift (d, t), we select the feasible employee with the smallest current hour balance. Let

$$F_{d,t}(x, xp) = \left\{e \in E \;\middle|\; LR\big(x \cup \{x_{e,d,t}{=}1\}, xp, E, D, T, V\big)\right\},$$

be the set of employees eligible for the shift, we choose the employee with the smallest hour balance

$$e^\star \in \arg\min_{e\in F_{d,t}(x,xp)} h_e, \quad \text{and set } x_{e^\star,d,t} \leftarrow 1,$$

updating $h_{e^\star} \leftarrow h_{e^\star} + h_t$ and the consecutive counter xp. If $F_{d,t}(x, xp) = \emptyset$, it is not possible to cover all shifts. The process is formalized in Algorithm 2.

Algorithm 2 Empty-shift backfill

Require: $x, xp, V, D_v, T_d, h_t, E, v_{cur}$

$U_v \leftarrow \{(d,t) : d \in D_{v_{cur}},\ t \in T_d,\ \sum_{e \in E} x_{e,d,t} = 0\}$
for all $(d,t) \in U_v$ **do**
    $F \leftarrow \{e \in E \mid \mathsf{LR}(x \cup \{x_{e,d,t}{=}1\}, xp, E, D, T, V)\}$
    if $F \neq \emptyset$ **then**
        $e^\star \leftarrow \arg\min_{e \in F} h_e$ ▷ break ties by selecting first employee
        $x_{e^\star,d,t} \leftarrow 1; \quad h_{e^\star} \leftarrow h_{e^\star} + h_t$
        $xp \leftarrow \texttt{updateConsecutive}(xp, e^\star, d, 1)$
    end if
end for

3.7.5 Local improvement

After Algorithm 1 and Algorithm 2 schedule shifts for a week, the resulting schedule may be suboptimal, as they are first-feasible algorithms. We therefore perform a local improvement using two algorithms: (i) relocation of a shift from one employee to another and (ii) swap of shifts between employees.

Objective. For week v, define employee hours

$$H_v(e) = \sum_{d \in D_v} \sum_{t \in T_d} h_t x_{e,d,t},$$

and minimize the deviation from weekly goals $g_{e,v}$ via

$$F_v(e) \;=\; \sum_{e \in E} |H_{e,v}(e) - g_{e,v}|,$$

Moves are accepted only if LR holds and $\Delta F_v < 0$.

Relocation (i). Relocation: pick an assigned shift (d,t) of e_{from} and move it to $e_{\text{to}} \neq e_{\text{from}}$:

$$x'_{e_{\text{from}},d,t} \leftarrow 0, \quad x'_{e_{\text{to}},d,t} \leftarrow 1,$$

if $LR(x, xp, E, D, T, V)$ holds and $F_v(x') < F_v(x)$. Defined in Algorithm 3.

Algorithm 3 Local improvement via relocation

Require: $x, xp, V, D_v, T_d, E, H_v, g_{e,v}, v_{cur}$

$\texttt{improved} \leftarrow$ **True**
while $\texttt{improved}$ **do**
    $\texttt{improved} \leftarrow$ **False**; $\quad (x^\star, \Delta^\star) \leftarrow (x, 0)$
    for all $d \in D_{v_{cur}}$ **do**
        for all $t \in T_d$ **do**
            $e_{\text{from}} \leftarrow \arg\max\{H_v(e) - g_{e,w}\}$
            for all $e_{\text{to}} \in E \setminus \{e_{\text{from}}\}$ **do**
                $x' \leftarrow x;\ x'_{e_{\text{from}},d,t} \leftarrow 0;\ x'_{e_{\text{to}},d,t} \leftarrow 1$
                if $LR(x', xp, E, D, T, V)$ **then**
                    $\Delta \leftarrow F_v(x') - F_v(x)$
                    if $\Delta < \Delta^\star$ **then**
                        $(x^\star, \Delta^\star) \leftarrow (x', \Delta)$
                    end if
                end if
            end for
        end for
    end for
    if $\Delta^\star < 0$ **then**
        $x \leftarrow x^\star$; $\texttt{improved} \leftarrow$ **True**
    end if
end while

Swap (ii). Swap: pick two assigned shifts from the same day (d, t) of e_1 and (d, t') of $e_2 \neq e_1$ and swap them:

$$x'_{e_1,d,t} \leftarrow 0, \quad x'_{e_2,d,t} \leftarrow 1, \qquad x'_{e_2,d',t'} \leftarrow 0, \quad x'_{e_1,d',t'} \leftarrow 1,$$

if $LR(x, xp, E, D, T, V)$ holds and $F_v(x') < F_v(x)$, defined in Algorithm 4.

Algorithm 4 Local improvement via swap

Require: $x, xp, D_v, T_d, E, g, v_{cur}$

```
improved ← True
while improved do
    improved ← False;   (x*, Δ*) ← (x, 0)
    for all d ∈ D_{v_cur} do
        for all t ∈ T_d, t' ∈ T_d do
            for all e1 ≠ e2 ∈ E with x_{e1,d,t} = 1 and x_{e2,d,t'} = 1 do
                x' ← x; x'_{e1,d,t} ← 0; x'_{e2,d,t} ← 1; x'_{e2,d,t'} ← 0; x'_{e1,d,t'} ← 1
                if LR(x', xp, ·) then
                    Δ ← F_v(x') − F_v(x)
                    if Δ < Δ* then
                        (x*, Δ*) ← (x', Δ)
                    end if
                end if
            end for
        end for
    end for
    if Δ* < 0 then
        x ← x*; improved ← True
        updateState(xp, x)            ▷ refresh xp and any hour balances from x
    end if
end while
```

3.7.6 Complete algorithm

The complete procedure for scheduling shifts for a contract structure $c \in C''$, producing a corresponding plan $P(c)$, is shown in Algorithm 5. The algorithm combines the defined components into one model. It should be noted that the shift allocation is run twice. First, with a smaller tolerance δ_1 to search for a more balanced solution for each employee, and then with a larger tolerance δ_2 to ensure that a feasible solution can be constructed for a sufficient number of employees.

Algorithm 5 Complete scheduling algorithm

Require: $C'', V, \delta_1, \delta_2$

for all $c \in C''$ **do**
  Assign winter vacations
  for all $v \in V$ **do**
    Identify employees eligible for annual leave
    Allocate shifts with δ_1
    Allocate shifts with δ_2
    Fill remaining empty shifts
    Apply local improvement
    Assign annual leave
  end for
end for
return $P(c), \forall c \in C''$

3.8 Realization phase

To evaluate the generated contract structures C'' and the corresponding plans $P(c)$ for $c \in C''$, we conduct simulations of employee behavior over the planning period. In these simulations, employees become sick and decide whether to accept offers to cover vacant shifts. This allows us to assess the flexibility of each contract structure.

Let E denote the set of employees in contract structure c. We let $S_{e,d} \in \{0, 1\}$ indicate whether an employee is sick, $S_{e,d} = 1$, or healthy ,$S_{e,d} = 0$. Furthermore, we let $VC_{e,d} \in \{0, 1\}$ indicate whether an employee is on vacation, $VC_{e,d} = 1$, or not, $VC_{e,d} = 0$. Combined, these variables describe the employee's state.

For each day d and shift t (with $d \in D$, $t \in T_d$), define the binary replacement variables

$$xr_{e,d,t} \in \{0, 1\} \quad \text{for } e \in E, \quad r_{d,t} \in \{0, 1\}, \quad ex_{d,t} \in \{0, 1\},$$

where $xr_{e,d,t} = 1$ indicates that employee e covers shift (d, t), $r_{d,t}$ indicates a rental employee covers shifts (d, t), and $ex_{d,t}$ indicates an external worker covers the shift (d, t). And the flag indicating a vacant shift,

$$vc_{d,t} \in \{0, 1\} \quad \text{for } e \in E,$$

with all replacement variables being constrained

$$\sum_{e \in E} xr_{e,d,t} + r_{d,t} + ex_{d,t} \leq vc_{d,t}, \quad \forall d \in D_v, \forall t \in T_d, \forall v \in V,$$

as a single absence can only be covered once. They describe the possible actions of an employee or external forces.

3.8.1 States

Vacation state $VC_{e,d}$. If an employee has either annual leave or a winter vacation, they can be considered equivalent for this model. Thus, the winter vacation allotment is preset as follows

$$VC_{e,d} \leftarrow 1, \quad d \in v_e \quad \forall e \in E,$$

and the annual leave

$$VC_{e,d} \leftarrow al_{e,d}, \quad \forall e \in E, \quad \forall d \in D_v \quad \forall v \in V,$$

moving employees into the vacation state.

Sick state $S_{e,d}$. Let employees fall sick on a given day d with the probability p_s, given that they are not already sick, with the length of the absence determined by the distribution S, with both parameters defined in Section 3.4. Let $R_{e,d} \geq 0$ define the number of remaining sick days. Thus, we can define the movement from the state of being healthy $S_{e,d} = 0$ as,

$$\Pr(S_{e,d} = 1 \mid S_{e,d-1} = 0) = p_s, \Pr(S_{e,d} = 0 \mid S_{e,d-1} = 0) = 1 - p_s,$$

and the movement from the state of being sick $S_{e,d} = 1$ as:

$$\Pr(S_{e,d} = 1 \mid S_{e,d-1} = 1) = \begin{cases} 1, & \text{if } R_{e,d-1} > 0 \\ 0, & \text{otherwise} \end{cases},$$

$$\Pr(S_{e,d} = 0 \mid S_{e,d-1} = 1) = \begin{cases} 0, & \text{if } R_{e,d-1} > 0 \\ 1, & \text{otherwise} \end{cases}.$$

When an employee becomes sick ($S_{e,d-1} = 0$ and $S_{e,d} = 1$), draw a duration $L_{e,d} \sim S$ and initialize the counter $R_{e,d} = L_{e,d} - 1$. Otherwise propagate

$$R_{e,d} = \begin{cases} \max\{R_{e,d-1} - 1, 0\}, & \text{if } S_{e,d-1} = 1, \\ 0, & \text{if } S_{e,d-1} = 0 \text{ and } S_{e,d} = 0. \end{cases}$$

If an employee is sick $S_{e,d} = 1$ on a given day d, the employee is unavailable for any shift he may have. Thus, the shift becomes vacant

$$vc_{d,t} = S_{e,d} * x_{e,d,t}, \quad \forall t \in T_d.$$

The employee movement between states is presented in Algorithm 6.

Algorithm 6 State update algorithm: Vacation $VC_{e,d}$ and Sickness $S_{e,d}$

Require: E, V, D_v, p_s, S, $x_{e,d,t}$,v_e, $al_{e,d}$

for all $v \in V$ **do**
    for all $d \in D_v$ **do**
        (1) Vacation state
        for all $e \in E$ **do**
            if $d \in v_e$ **then** ▷ winter vacation days
                $VC_{e,d} \leftarrow 1$
            else
                $VC_{e,d} \leftarrow al_{e,d}$ ▷ annual leave indicator for day d
            end if
        end for

        (2) Sickness process
        for all $e \in E$ **do**
            if $S_{e,d-1} = 0$ **then** ▷ was healthy yesterday
                Employees become sick with probability p_s (if not already sick)
                if *becomes sick* **then**
                    $S_{e,d} \leftarrow 1$
                    draw $L_{e,d} \sim S$ ▷ sick-leave length
                    $R_{e,d} \leftarrow L_{e,d} - 1$
                else
                    $S_{e,d} \leftarrow 0, \quad R_{e,d} \leftarrow 0$
                end if
            else ▷ $S_{e,d-1} = 1$ was sick yesterday
                if $R_{e,d-1} > 0$ **then**
                    $S_{e,d} \leftarrow 1, \quad R_{e,d} \leftarrow R_{e,d-1} - 1$
                else ▷ $R_{e,d-1} = 0$
                    $S_{e,d} \leftarrow 0, \quad R_{e,d} \leftarrow 0$
                end if
            end if
        end for

        (3) Vacancies
        for all $t \in T_d$ **do**
            $v_{d,t} \leftarrow \sum_{e \in E} x_{e,d,t} S_{e,d}$
        end for
    end for
end for

3.8.2 Actions

Given a vacant shift $v_{d,t} = 1$ on day d, we consider two types of actors who can attempt to cover it:

- **Internal employees:** employees not at work, on vacation, or sick,

$$E' = \{e \in E \mid VC_{e,d} = 0,\ S_{e,d} = 0,\ x_{e,d,t} = 0\},$$

- **External actors:** employees from other branches and rental workers.

Our set of internal employees act as individual actors, and thus each internal employee has a probability $p_{c,w}$ of accepting a shift. External actors are considered a pool of an unknown size. Thus, we define the probability as the chance that the shift is filled by an external entity. Given that the percentage of shifts filled by a rental worker is p_r, the percentage of shifts filled by an employee from another branch is p_{ab}, and the percentage of shifts filled by internal employees p_f, we define p_{rental}, the probability a shift is filled by a rental worker, in Equation (17), and $p_{another}$, the probability a shift is replaced by an external employee, in Equation (16). These probabilities reflect the likelihood that a vacancy unable to be covered by internal employees is distributed to either a rental worker, external employee, or remains vacant. Shifts are offered sequentially: first to internal employees, then to employees from other branches, and finally to rental workers. Internal employees are offered shifts in random order.

$$p_{another} = \frac{p_{ab}}{p_{nf} + p_r + p_{ab}}. \tag{16}$$

$$p_{rental} = \frac{p_r}{p_{nf} + p_r + p_{ab}}, \tag{17}$$

Before offering shifts, LR constraints must be satisfied. The rules governing additional voluntary shifts are less strict than those in the planning phase:

1. **Consecutive shift limit.** For each employee e and day d,

$$xp_{e,d} \ \le\ m_r = 10, \qquad \forall e \in E,\ \forall d \in D_v,\ \forall v \in V.$$

2. **Total hours limit.** Over the planning horizon,

$$\sum_{v\in V} \sum_{d\in D_v} \sum_{t\in T_d} h_t(x_{e,d,t} + xr_{e,d,t}) \ \le\ h_{\max}, \qquad \forall e \in E,$$

   where $h_{\max} = 37.5 \cdot |V|$.

3. **Weekly hours limit.** For each week $v \in V$,

$$\sum_{d\in D_v} \sum_{t\in T_d} h_t(x_{e,d,t} + xr_{e,d,t}) \ \le\ h_{\text{week}} = 48, \qquad \forall e \in E,\ \forall v \in V.$$

4. **Daily shift limit.** For each employee e and day d,

$$\sum_{t \in T_d} \left(x_{e,d,t} + xr_{e,d,t} \right) = 1, \qquad \forall e \in E,\ \forall d \in D_v,\ \forall v \in V.$$

We re-utilize the convention from the planning phase by defining the function $LR_r(e)$, which evaluates whether employee e remains within LR limits if assigned an additional voluntary shift. The feasible set of internal employees is

$$E'' = \{e \in E' \mid LR_r(e) = 1\}.$$

Replacement process. Given a vacant shift $v_{d,t} = 1$, we iterate over employees in a random order of E''. Each employee e is offered the shift with acceptance probability $p_{c,w}$, conditional on no earlier employee having accepted:

$$\Pr\left(xr_{e,d,t} = 1 \middle| v_{d,t} = 1,\ xr_{e',d,t} = 0\ \forall e' \neq e \right) = p_{c,w}.$$

If an employee accepts, we update

$$xr_{e,d,t} \leftarrow 1, \qquad xp_{e,d} \leftarrow \textbf{updateConsecutive}(xp, e, d, 1).$$

If no internal employee accepts, we attempt to assign an employee from another branch with probability $p_{another}$, updating

$$ex_{d,t} \leftarrow 1 \quad \text{if successful.}$$

If no external employee accepts ($ex_{d,t} = 0$), we attempt to assign a rental worker with probability p_{rental}, updating

$$r_{d,t} \leftarrow 1 \quad \text{if successful.}$$

Finally, the realized status of the shift is given by

$$v_{d,t} = \sum_{e \in E''} xr_{e,d,t} + ex_{d,t} + r_{d,t}.$$

In Algorithm 7, we provide a summary of the acceptance process for a given vacancy.

Algorithm 7 Cover a vacant shift (d, t)

Require: $v_{d,t} = 1$,E, V, $\{D_v\}$, $VC_{e,d}$, $S_{e,d}$,$x_{e,d,t}$,$p_{c,w}$ (internal), $p_{another}$, $p_r ental$, $LR_r(e)$

Candidate sets
$E' \leftarrow \{e \in E \mid VC_{e,d} = 0,\ S_{e,d} = 0,\ x_{e,d,t} = 0\}$ ▹ internal, available
$E'' \leftarrow \{e \in E' \mid LR_r(e) = 1\}$ ▹ LR-feasible employees

(1): Internal employees
Offer shift to employees in E'' in random order
Each employee $e \in E''$ accepts with probability $p_{c,w}$, provided no earlier employee has accepted
if an employee e accepts **then**
    $xr_{e,d,t} \leftarrow 1$
    $xp_{e,d} \leftarrow$ **updateConsecutive**$(xp, e, d, 1)$
    return shift covered
end if

(2): External employee
If no internal employee accepted:
Offer shift to external employees, acceptance occurs with probability $p_{another}$
if accepted **then**
    $ex_{d,t} \leftarrow 1$
    return shift covered
end if

(3): Rental workers
If no external employee accepted:
Offer shift to rental workers, acceptance occurs with probability p_{rental}
if accepted **then**
    $r_{d,t} \leftarrow 1$
    return shift covered
end if

// Final status
If no actor accepts, $v_{d,t}$ remains vacant

3.8.3 Simulations

Given the agent behavior, states, and actions defined above, we can simulate the flexibility of each contract structure over the planning period. Because both sickness and replacement processes are stochastic, the outcome of a single simulation run is not representative. To obtain accurate estimates, we therefore adopt a Monte Carlo approach [15]. By repeating the simulation a sufficiently large number of times, M, the average of the performance measures converges to their expected values.

Let $\mathrm{SIM}(c) = \{sim_1, sim_2, \dots, sim_m\}$ denote the set of m independent simulation replications of a contract structure $c \in C''$. Each replication sim_j generates a realization of employee states and actions over the planning period. The overall simulation process for a given contract structure c can therefore be defined as

$$\mathrm{SIM}(c) = \big\{sim_j(c; \xi_j) \bigm| j = 1, \dots, m\big\},$$

where ξ_j denotes the random seed.

3.9 Evaluation phase

When evaluating the contracts, we have three primary key performance indicators: viability, cost, and flexibility. Our viability metric is defined as all shifts being filled at the planning phase. We reject contracts from further study if they fail to schedule all shifts, as it is the most basic requirement of the business. We can formalize the requirement as

$$\sum_{e \in E} x_{e,d,t} = 1, \qquad \forall t \in T_d \quad \forall d \in D_v \quad \forall v \in V.$$

The personnel costs of a business are complex, including different factors such as shift type and role. However, we can simplify the calculations for our needs. As demand is fixed, costs related to planned shifts can be excluded, as they remain constant between contract structures. We further exclude extra costs generated by employees accepting vacant shifts, as the goal of the business is to fill every vacancy. Costs we do consider include the fixed costs of each employee, such as insurance, work clothes, and licenses. For this, we estimate an annual cost of 2000€, denoted as r_e. Furthermore, we must consider the possibility of undertime occurring during the planning phase. Given an estimated hourly rate of 25€, denoted as r_h, an employee is entitled to pay for all hours specified in their contract, even if they cannot be scheduled for them. Noteworthy here is that any additional shifts taken during the realization phase are not included. It should be emphasized that the numerical cost parameters used in the analysis are the author's own estimates. They are included to describe the methodology, and decision-makers should replace them with more accurate values. The cost can be formulated for a given contract $c \in C''$ as follows:

$$Cost(c) = N(c) r_e + \sum_{e \in E} \max\{ch_e - \sum_{v \in V} \sum_{d \in D_v} \sum_{t \in T_d} h_t x_{e,d,t}, 0\} r_h.$$

Finally, we must consider flexibility, which we define as the ability of the contract structure to respond to vacancies. We formalize the acceptance rate of one simulation as

$$\alpha(sim) = \frac{\sum_{d \in D} \sum_{t \in T_d} vd_{d,t}}{\sum_{d \in D} \sum_{t \in T_d} vc_{d,t}}.$$

Thus, we can formalize the flexibility of the given contract c as the Monte Carlo average

$$F(c) = \frac{1}{|\,\mathrm{SIM}\,|} \sum_{sim \in \mathrm{SIM}} \alpha(sim).$$

3.10 Tools used

The simulation model was implemented in Python 3.10 [20] using several packages. The model logic and computations were developed with NumPy [8], Pandas [13], and SciPy [23]. For visualization and exploratory analysis, Matplotlib's Pyplot interface [9], Plotly [18], and Seaborn [24] were utilized. All computational models were developed internally using these tools.

4 Computational results

This section presents the results of applying the simulation model to a single location from the dataset. The businesses are left anonymous. The analysis covers a three-month planning period from January 6 to April 6, 2025, while parameter estimates are based on one year of historical data (June 1, 2024, to June 1, 2025).

The example location has the contract structure shown in Table 3, recorded on the first day of the planning period. In total, the location has 417,5 contracted hours. Over the 13 weeks, the total workload amounts to 4,854.75 hours, with an average shift length of 6.78 hours and an average weekly demand of 373.44 hours. Furthermore, we estimate that 80 percent of employees will take one week of winter vacation during this period, as not all employees will have accumulated enough vacation days.

Table 3: Contract structure of the example location

Contract type	Nr of employees
37.5	3
35	1
30	6
20	3
10	2
5	2

4.1 Parameter estimations

To initialize the simulation model, we first estimated the key parameters from historical data. These include the probability of an employee falling ill p_s, the proportion of shifts covered by sickness p_{sick}, the distribution of sick leave lengths S, the acceptance rate of vacant shift offers for internal employees $p_{c,w}$, and the proportion of vacant shifts covered by rental employees p_r, and external employees p_{ab}.

Figure 2 shows the distribution of sick leave lengths S. A key takeaway is that the average sick leave is short, with a median length of two days and an average length of 2.4 days. This result is expected, as most sick leaves are short, typically due to issues such as a cold or stomach problems. Furthermore, employees may take up to three days of sick leave without a doctor's note, which may impact employee decision-making.

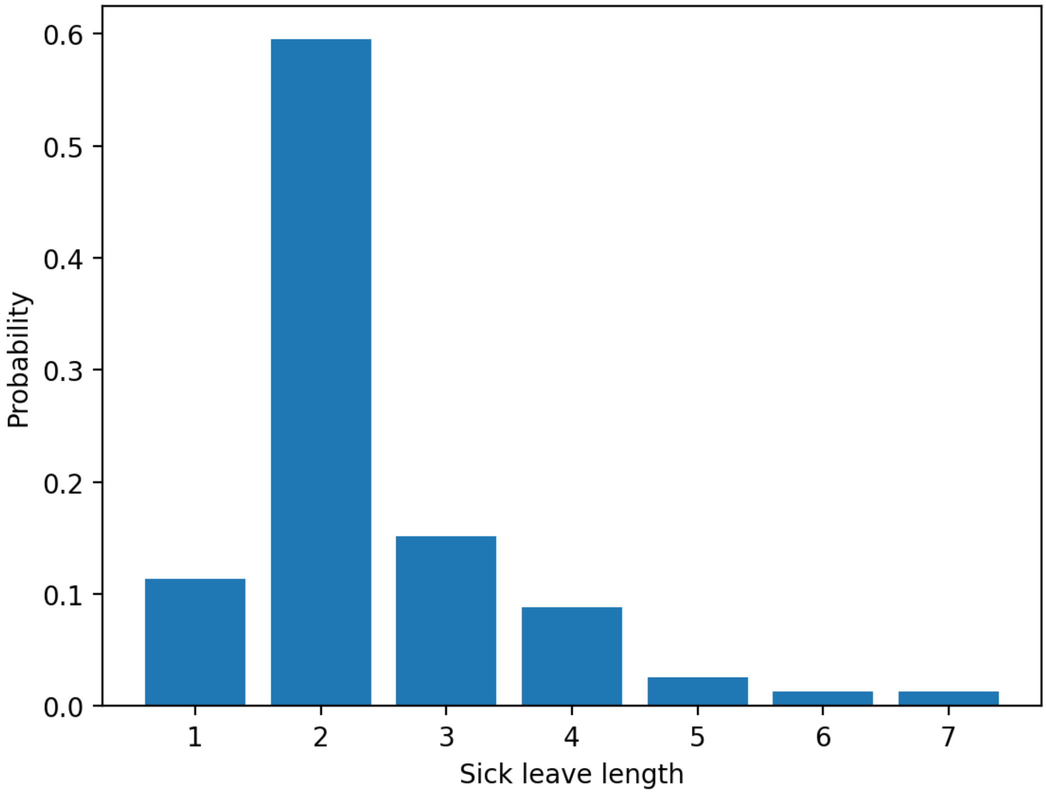

Figure 2: The distribution of sick leave lengths.

Table 4 shows the probability of an employee falling sick, a rental employee accepting an additional shift, and an external employee accepting an additional shift. Notably, rental workers and external employees are not utilized at a high rate. We note that the proportion of shifts covered by sickness is relatively low, with a corresponding daily sickness rate of 1.62 percent. In practice, this means that an employee gets sick about $365 * 0.0162 \simeq 6$ times per year.

Table 4: Estimated probabilities for sickness and shift acceptance, excluding employees working at the location.

Parameter	Probability
Proportion of shifts covered by sickness (p_{sick})	3.89%
Daily probability of employee falling sick (p_s)	1.62%
Percentage of vacant shifts accepted by internal employees (p_f)	58.33%
Percentage of vacant shifts accepted by external employees (p_{ab})	5.56%
Percentage of vacant shifts accepted by rental employees (p_r)	5.21%

Table 5 presents the probability that an employee accepts a shift on a given day, depending on both the weekday and the employee's contracted hours. Contrary to the common assumption that employees with small contracts ($\leq$ 10 h) are more likely to accept additional shifts, the results show that employees with larger contracts ($\geq$ 10 h) are generally more likely. Interestingly, all groups except those working 11–19 hours per week show higher acceptance rates on Sundays, partially supporting our hypothesis that Sunday vacancies are easier to fill due to higher pay. Overall, acceptance rates are highest among employees with mid-sized contracts (11–29 h).

Table 5: Estimated probability of an employee accepting a shift. Rows indicate weekday groups, while columns represent contracted weekly hours. Table values correspond to parameter p_{cw}, the probability that an employee accepts an additional shift on a given day.

	0–10 h	11–19 h	20–29 h	30–36.5 h	37.5 h
Mon–Thu	5.5%	14.2%	11.6%	11.6%	7.1%
Fri–Sat	10%	20.7%	13.4%	6.8%	5.5%
Sun	13.0%	4.9%	14.1%	14.1%	4.2%

4.2 Contract generation phase

We base our goal hours for the contract generation phase on the average weekly demand, defined $D_{avg} = 415.93$ hours per week. In addition, we account for contract hours allocated to winter vacations and annual leave. Since 80 percent of employees are expected to take one week of winter vacation, the weekly adjustment is given by

$$D_{wv} = \frac{D_{avg}}{|W|} * 0.8,$$

where W is the set of weeks in the planning horizon.Annual leave accounts for approximately $7.5/200 = 3.75\%$ of contract hours. The corresponding adjustment is therefore

$$D_{an} = D_{avg} * (0.0375).$$

Industry practice is that total contract hours should be set below demand, with 95 percent considered a suitable benchmark. This buffer reduces the risk of undertime, since part-time employees can be scheduled for slightly more hours than their contract hours, providing planners with additional flexibility. The total effective demand is therefore defined as

$$D_{tot} = 0.95 * (D_{avg} + D_{an} + D_{wv}).$$

The algorithm requires as input a set of possible contract types $T = \{37.5, 30, 20, 15, 5\}$. We restrict the total number of employees to lie within bounds $N_{min}, N_{max} = 16, 20$, and fix the number of full-time employees to the current level, reflecting their specialized roles in the business: $L_{min}(37.5), L_{max}(37.5) = 3, 3$. Finally, we set $\delta = 0.05$, allowing the algorithm defined in Section 3.6 to generate 16 unique contract structures, C'', in addition to the current structure. The complete set of 17 structures is shown in Figure 3, where cs denotes the current contract structure and $c_i, i \in [1, 16]$, the generated contract structures. The numbers shown next to each identifier correspond to the number of employees in that contract structure.

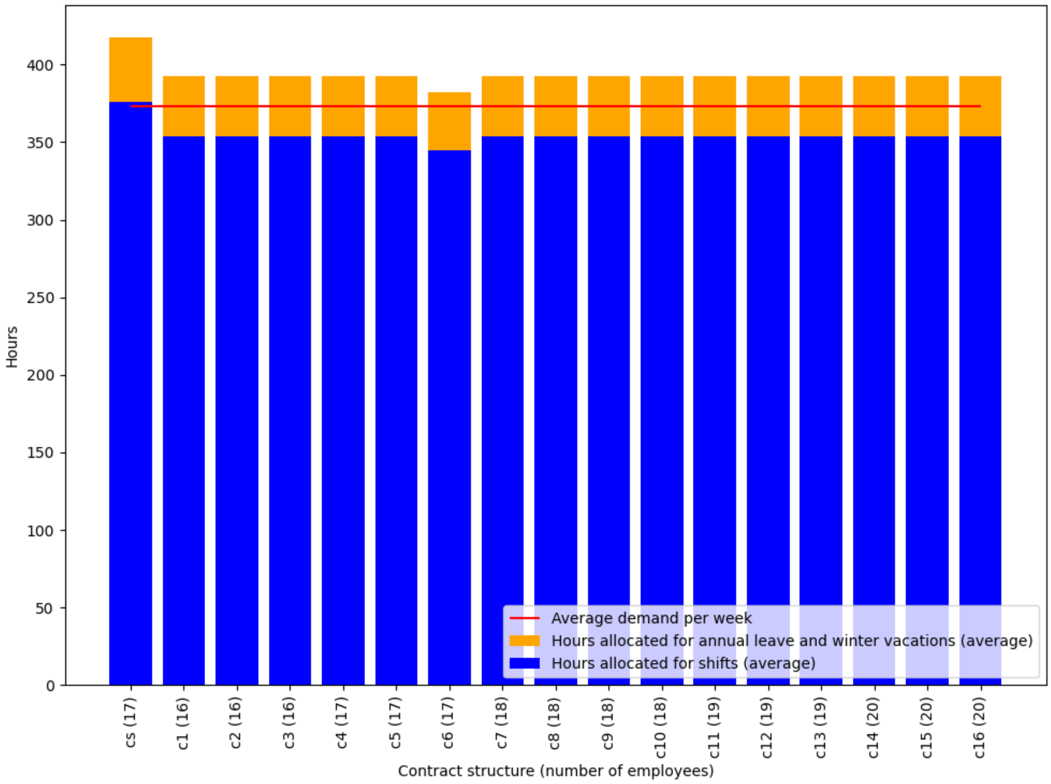

Figure 3: The generated set of contract structures.

We observe that the algorithm successfully generates feasible contract structures. Although the total contract hours exceed the estimated demand, the hours allocated towards shifts correspond to the expected 95 percent. The full set of generated contract structures is provided in Table A1 in the appendix.

4.3 Planning phase

The planning phase involves scheduling employees based on a set of input parameters. Specifically, we incorporate the previously defined vacation allotment of 80 percent, the set of planned shifts T_p, the set of contract structures C'', and the tolerance parameters $\delta_1 = 1.5$ and $\delta_2 = 7$. Using the allocation algorithm described in Section 3.7, we generate a set of feasible plans

$$P(c), c \in C'',$$

where each $P(c)$ represents the plan for contract structure c.

When evaluating the validity of the plans, we must consider three factors: whether it was possible to formulate a plan covering all shifts, while adhering to the LR requirements, whether undertime was required, and how evenly hours were distributed among employees from week to week.

After distributing the shifts, we find that a feasible plan covering all shifts could be created for each of the generated contract structures. To verify that the model correctly rejects unfit contract structures, we conducted a stress test using a structure consisting solely of full-time employees. As expected, this configuration failed, confirming that the allocation algorithm checks for viability.

4.3.1 Observed under- and overtime

In Figure 4, we report observed under- and overtime across contract structures. As expected, there is a substantial amount of overtime. Given our design target, setting total contracted hours to approximately 95% of expected demand, overtime is intentional and functions as a feature. It eases planning and follows real-world practices.

We observe that only the current contract structure generates substantial amounts of undertime. This outcome is logical, as the total contracted hours exceed the expected demand, as shown in Figure 3. This finding further supports the accuracy of our demand estimates used in generating the contract structures.

We also observe small amounts of undertime in several contract structures. These cases stem from the allocation algorithm under-scheduling full-time employees near the end of the planning period, due to overtime not being allowed for full-time employees and variance in shift length. For example, suppose a full-time employee requires 36.2 hours in the final week to meet the contract hours, but the remaining feasible shifts allow at most 36.0 hours. In that case, a residual 0.2 hours of undertime remains. This effect is not observed for other contract types, which are more flexible. Given that the magnitudes are small and stem from the algorithm's inability to look ahead rather than the contract structures themselves, these instances are acceptable. This is further visualized and explored in Section 4.3.2.

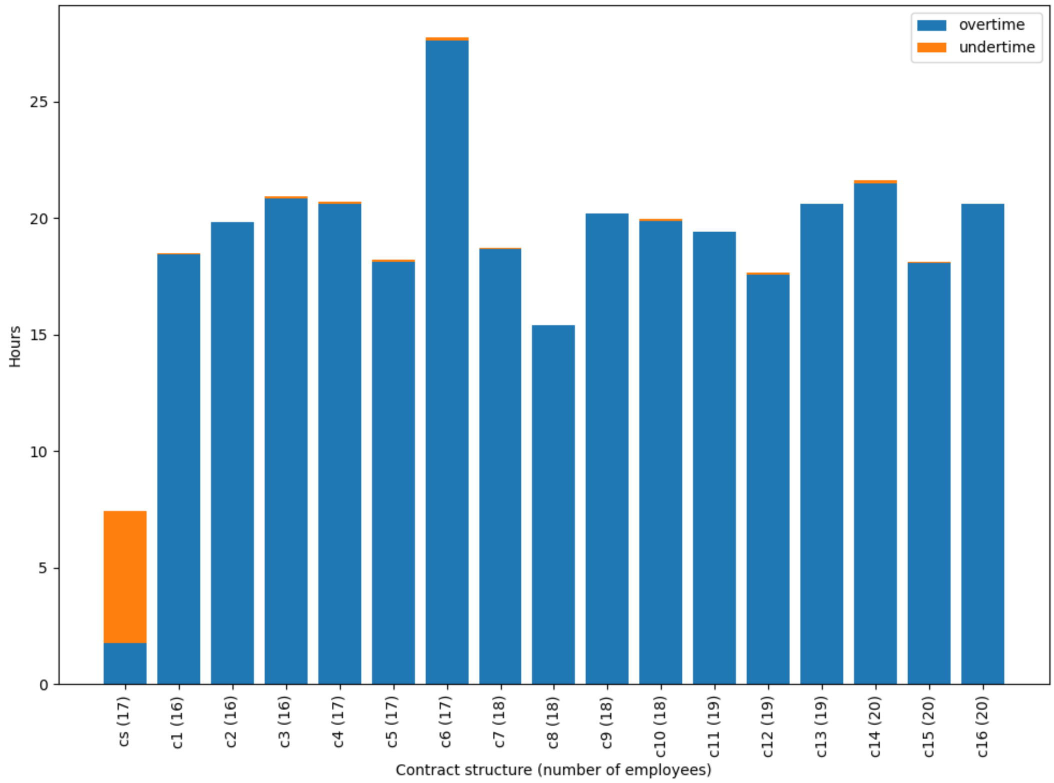

Figure 4: Observed under- and overtime in scheduling.

4.3.2 Distribution of employee hours

A key metric for evaluating the efficacy of a contract structure is the deviation between employees' planned working hours and their contracted hours at the end of the planning period. While part-time employees are expected to work some overtime, the amount should remain small and evenly distributed. If an employee consistently works substantial overtime during the planning period, their contracted hours should be adjusted accordingly.

Figure 5 presents a scatter plot comparing total planned hours and contracted hours, with full-time and part-time employees shown separately for clarity. The results indicate that overtime among part-time employees is evenly distributed across all contract structures. A maximum deviation of 30 hours is considered acceptable, as it corresponds to less than three hours per week on average. If the decision-maker desires less systematic overtime, the percentage used in the generation algorithm can be increased. Furthermore, full-time employees do not exhibit any overtime, as intended. However, some full-time employees exhibit very minor undertime, highlighting the issue discussed in Section 4.3.1.

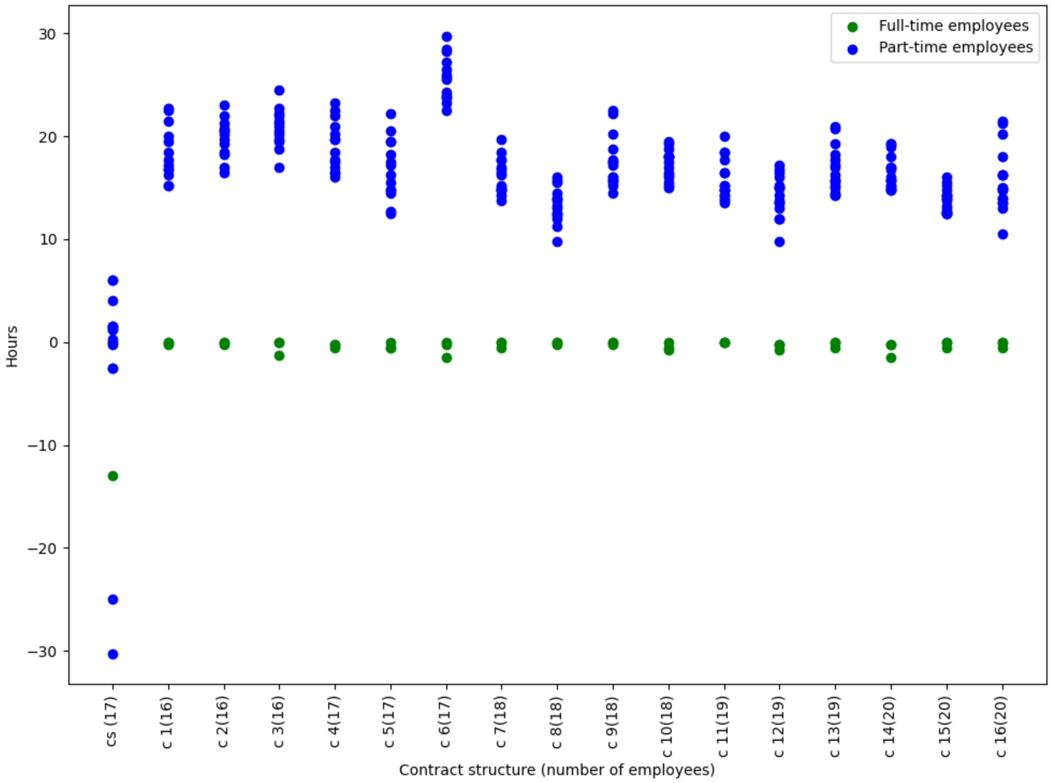

Figure 5: The total difference between contract hours and total hours for each contract structure, for each employee.

4.3.3 Variance in scheduling

To quantify how evenly shifts are distributed, we define the weekly relative workload

$$H_{e,w} = \sum_{d \in D_w} \sum_{t \in T_d} h_t x_{e,d,t}, \qquad r_{e,w} = \frac{H_{e,w}}{ch_e}, \quad e \in E, \ w \in W,$$

where h_t is the length of shift t and $x_{e,d,t} \in \{0, 1\}$ indicates whether employee e is assigned to that shift. Thus $r_{e,w} = 1$ means the employee was scheduled exactly to their contract hours, $r_{e,w} > 1$ overscheduled, and $r_{e,w} < 1$ underscheduled. As our contract hours should by design not equal demand, the mean value of $r_{e,w}$ will be greater than 1. By measuring the standard deviation on $r_{e,w}$, we define a metric that considers the variance in hours done from week to week. In Figure 6 we compare them. Here we can note that the standard deviation ranges from ~ 0.12 up to ~ 0.35, all considered acceptable values.

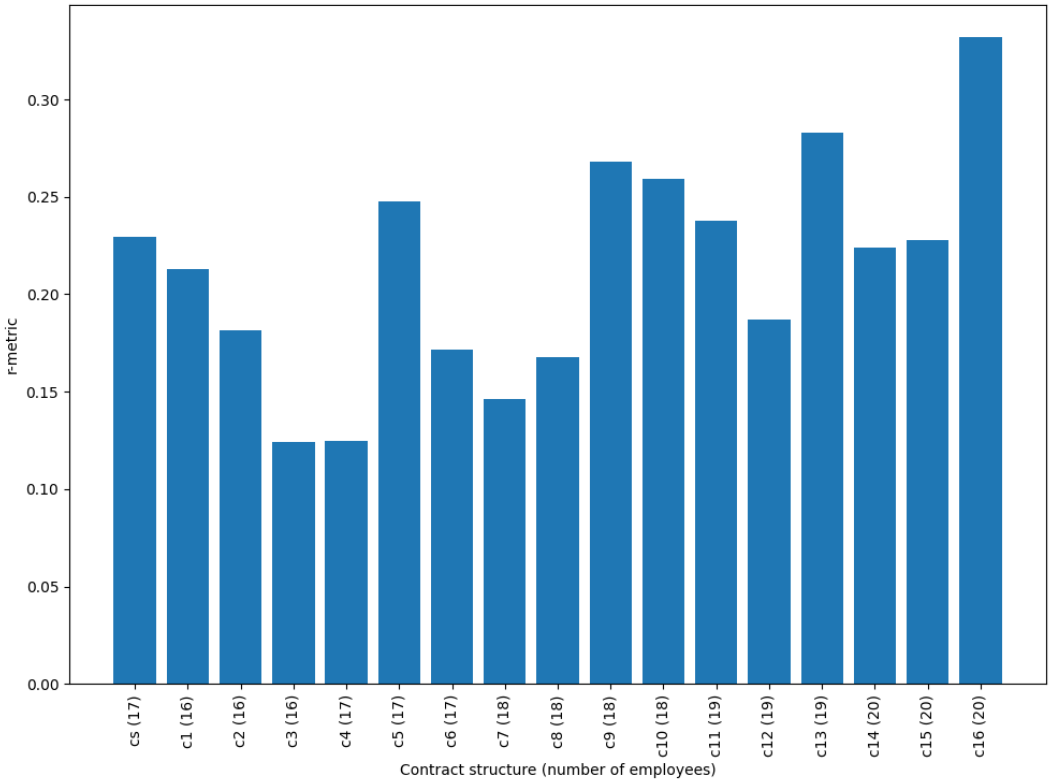

Figure 6: The standard deviation of $r_{e,w}$ values for each contract structure.

Figure 7 shows a box plot of the $r_{e,w}$ values for each contract structure considered. We note that the spread tends to be tighter in structures with fewer employees. Furthermore, the frequency of values greater than two is small, especially outside the current structure.

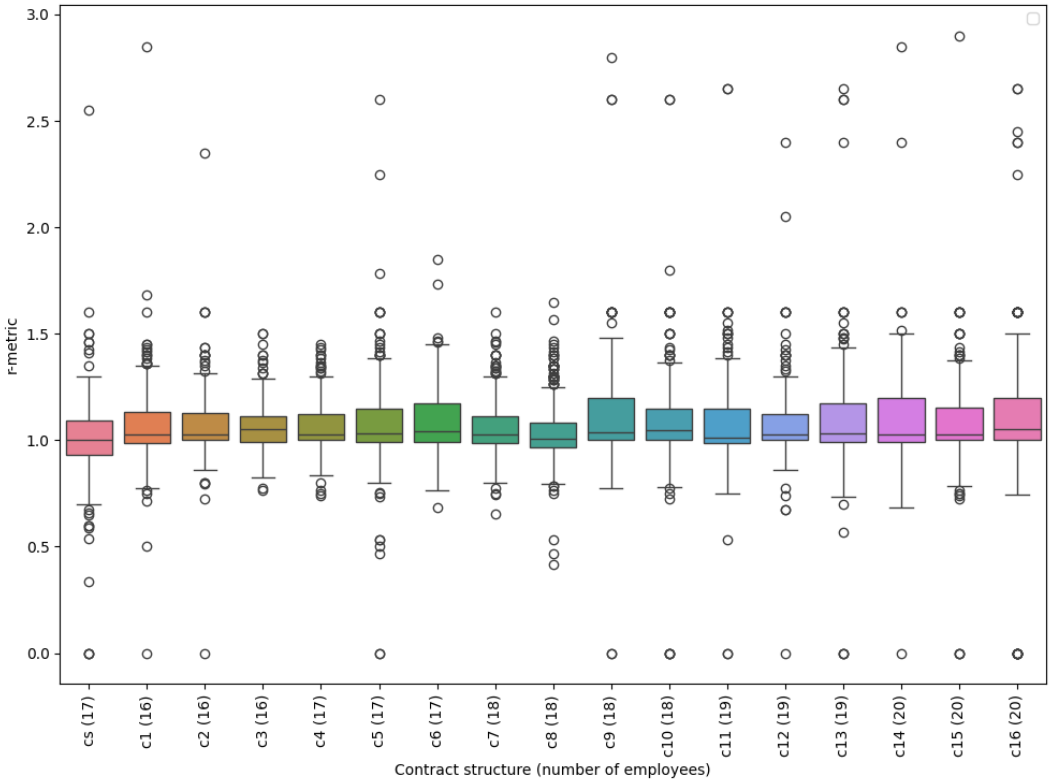

Figure 7: Box plot of $r_{e,w}$ values per contract structure.

Figure 8 shows a box plot of the $r_{e,w}$ values divided by contract type. Here, we note that the spread is tighter for larger contract types. This shows our algorithm behaves as intended. The algorithm schedules larger contracts first, meaning they are the beneficiaries of tie-breaks. Furthermore, it calculates the absolute difference between the hours scheduled and the contract hours. Thus, the relative difference $r_{e,w}$ will be larger for smaller contracts. It should be noted that contract types only present in the current contract structure were removed due to the small sample size.

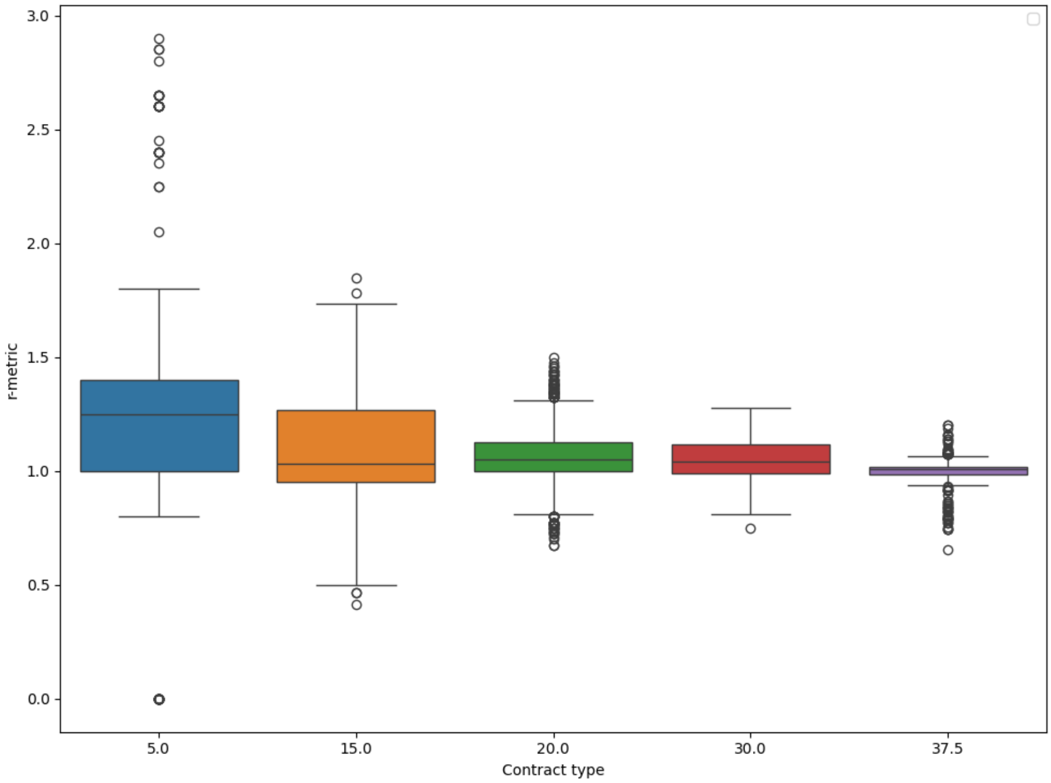

Figure 8: Box plot of $r_{e,w}$ values per contract type.

4.3.4 Validation of labor regulations

The algorithm must adhere to all LR constraints specified in Section 3.7.2. Correct behavior is verified by checking the maximum observed value for each constraint. This approach ensures that no employee violates any constraint, provided the maximum value is equal to or below the defined limit. The corresponding constraint values are summarized in Table 6. Notably, all constraints, except the weekly hours limit, reach their respective maximum values.

Table 6: Maximum observed values for LR constraints in the planning phase.

Constraint	Maximum value observed
Weekly shift limit	5
Consecutive shift limit	8
Total hours limit	487.5
Weekly hours limit	45
Daily shift limit	1

4.3.5 Example result from single contract

Figure 9 shows the distribution of working hours in contract structure c_1. We observe that 12 out of 16 employees (75%) take a winter vacation week. In addition, all employees receive varying amounts of annual leave. Because of how annual leave is allocated in the model, some annual leave is expected to carry over to the next planning period. In this case, three employees have accumulated enough hours for an additional day of leave that remains unused.

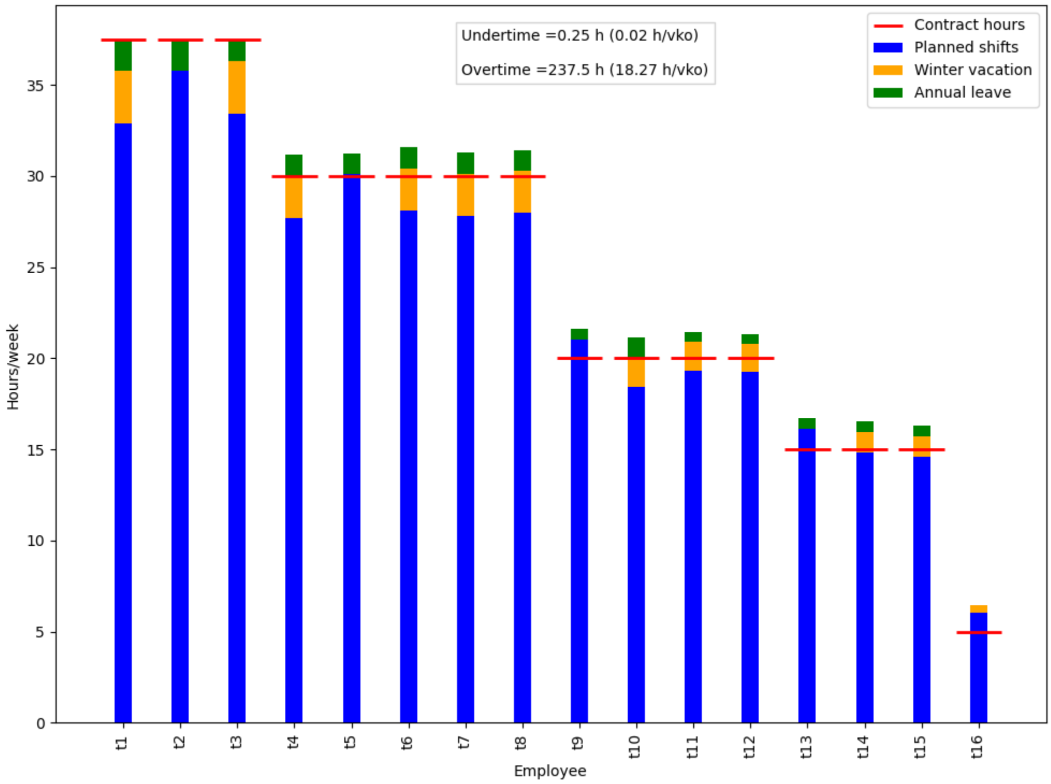

Figure 9: Average hour usage in a week, plotted for each employee in contract structure 1 (c1).

4.3.6 Summary

We conclude that the model provides an accurate representation of the planning phase as defined within the scope of this thesis. Although variance exists in scheduled overtime among employees and in weekly working hours, the overall performance is satisfactory. All predefined hard constraints are satisfied. This confirms that the model generates feasible and fair schedules. This allows us to use the schedules as a basis for the realization and evaluation phases.

4.4 Realization phase

This subsection evaluates how the contract structures C'' perform under uncertainty. Using the plans from the planning phase as a base, we simulate the three-month horizon (6 January–6 April 2025) with the empirically estimated probabilities for sickness, and shift acceptance (employees, rental, external). For each contract structure $c \in C''$, we run 1000 independent replications.

4.4.1 Illustrative results

Figure 10 plots the average rate of vacant shifts against the number of employees in each contract structure. The results show a negative correlation between the number of employees and the vacancy rate, as indicated by the trend line. However, the in-group variance is high, suggesting that the number of employees alone does not account for the differences between contract structures. Overall, these findings are consistent with our expectations.

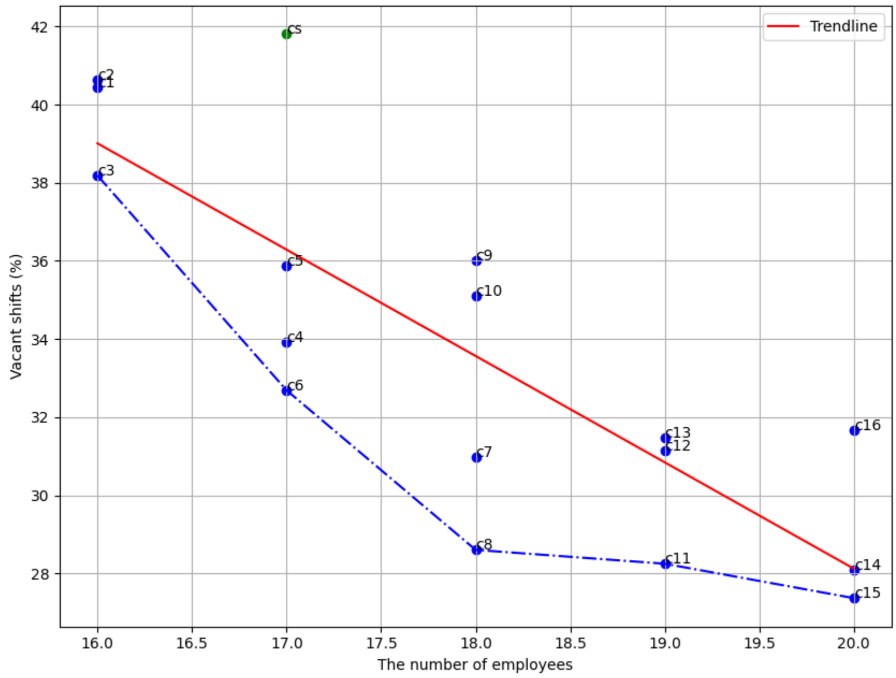

Figure 10: Plot comparing the rate of vacant shifts and the number of employees for the set of contract sturctures.

It is noteworthy that the proportion of vacant shifts remains relatively high. However, given that the empirically observed vacancy rate is approximately 30%, the results are not unrealistic. The estimated employee willingness to accept vacant shifts is based on data from the entire organization, so some deviation for a single site is expected. Furthermore, the model restricts full-time employees from covering vacant shifts due to forbidding paid overtime, whereas paid overtime is sometimes allowed in real-world operations.

Figure 11 illustrates how vacant shifts were replaced. The results show that most replacements were done by internal employees. The replacement rates for external employees and rental workers remain relatively constant, although a slight increase is observed as the number of vacancies not covered by internal employees grows. Overall, the proportions align with the expected values, though the rates of rental and external employee usage are slightly higher than expected.

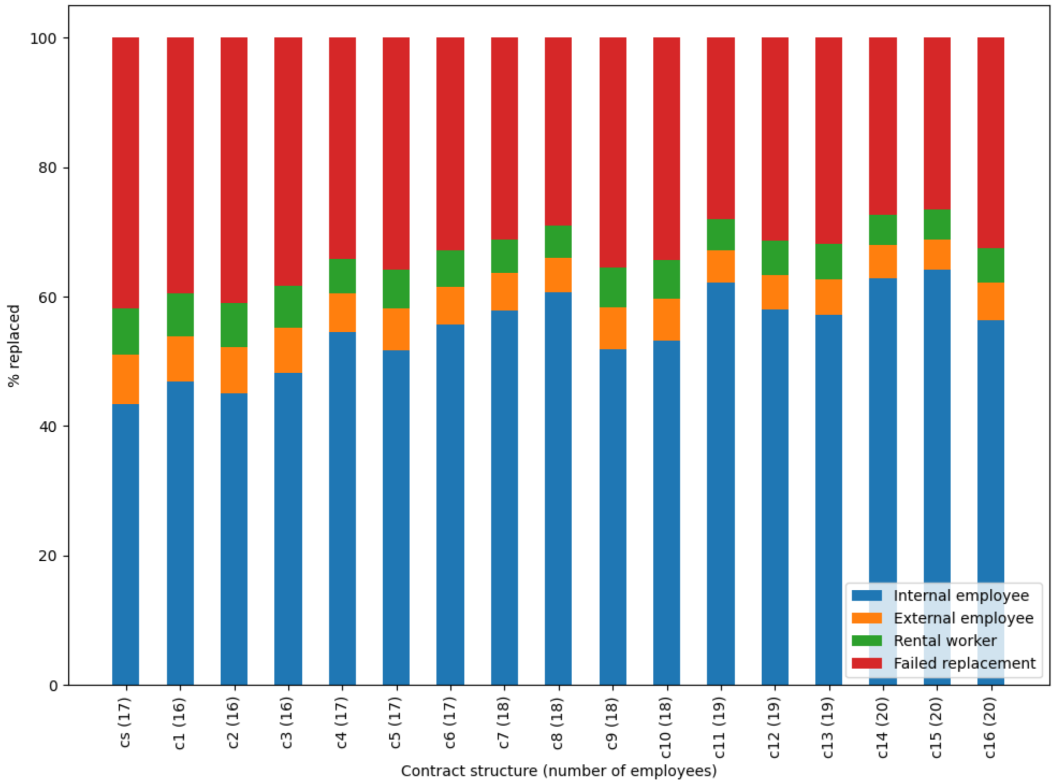

Figure 11: Histogram on the treatment of vacancies in each contract structure.

Figure 12 presents a box plot where each data point represents the average number of excess hours an employee worked per week in the simulation, grouped by contract type. It should be noted that only one employee in the current contract structure works 35 hours per week, and none do so in the generated structures. Thus, the sample size for this group is small, and no meaningful conclusions can be drawn from that employee's behavior in the model. Similarly, the current contract structure includes two employees working 10 hours per week, who are absent in the generated contract structures and should therefore be excluded from the comparison. The figure indicates that employees with mid-sized contracts (15 hours per week) perform the most replacements.

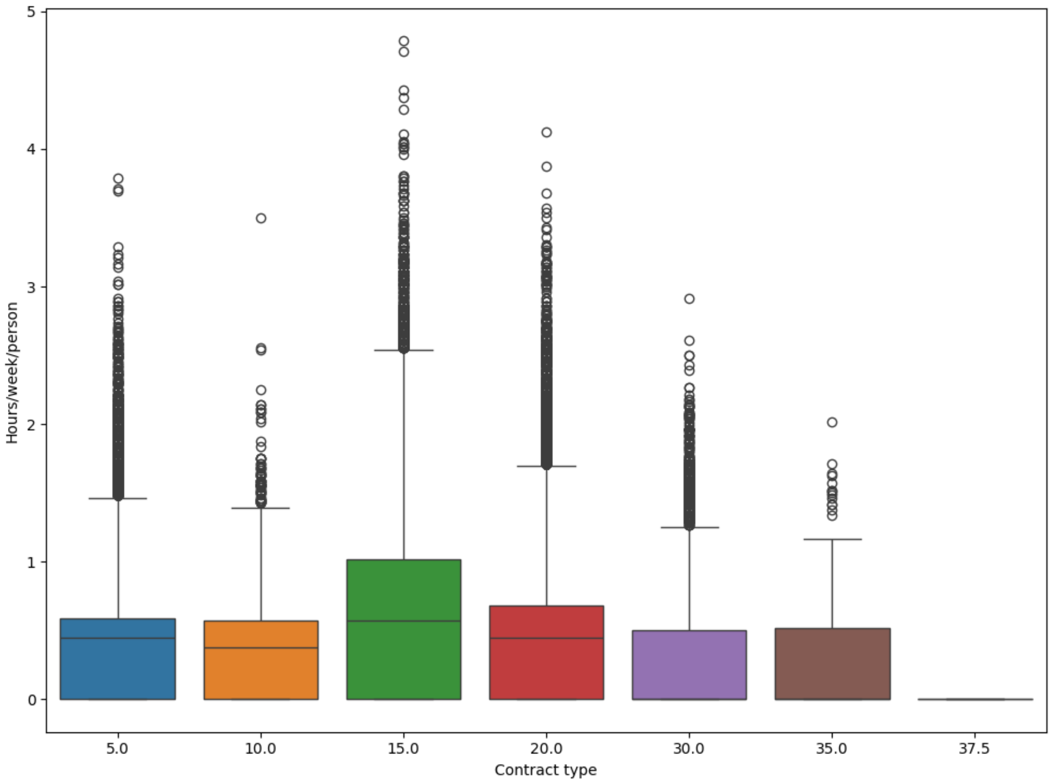

Figure 12: Box-plot of excess hours performed in each simulation per contract type.

We examine the distributions of four key metrics: the number of sick leaves and replacement shares by internal employees, external employees, and rental workers. We examine the behavior of the metrics and detect deviations from our expectations, which are that the metrics are normally distributed. We run 10,000 replications to ensure the behavior is not affected by the random seed. Figure 13 shows the empirical distributions. The number of sick employees and the share of replacements by internal employees do seem to be normally distributed.

However, the replacement rates of rental and external workers are right-skewed. This is due to low base rates with frequent zeros, simulations without such replacements. Consequently, the median replacement rate aligns closely with the parameter value, while the mean is higher. Because these replacements cover a small share of vacancies, the deviation is minor.

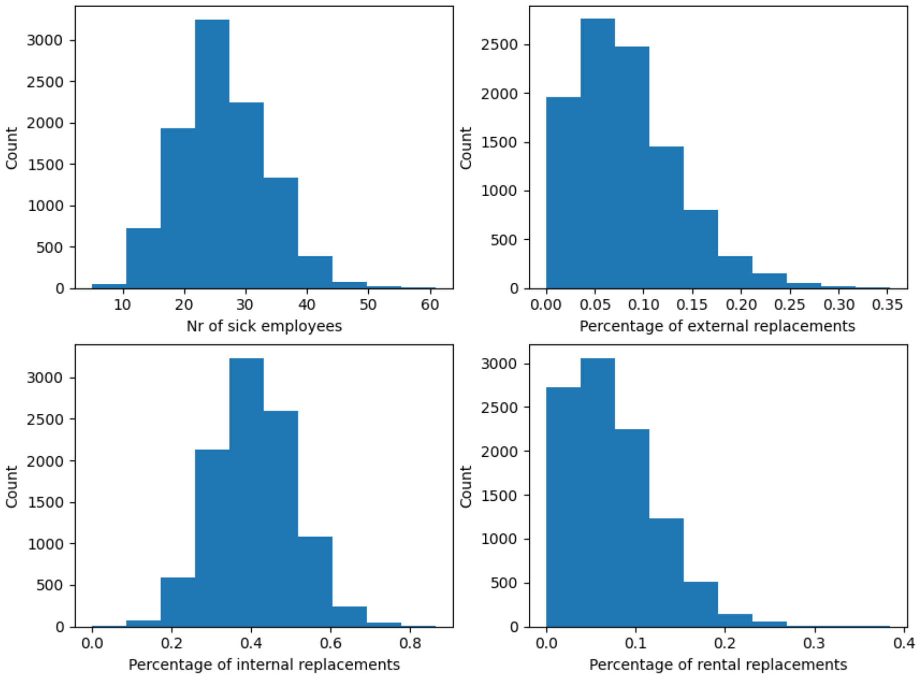

Figure 13: Distributions of key metrics.

4.4.2 Validation

To ensure the model behaves correctly, we validate its behavior using three methods: LR requirements, extreme-case behavior, and reproducibility. First, we verify that all schedules produced by the simulator satisfy the LR requirements, modeled as hard constraints. Second, we confirm the model behaves correctly under extreme inputs (such as zero sicknesses, guaranteed replacements), where outcomes are predictable. Third, for a fixed set of inputs, we require outcomes to be consistent with different random seeds.

Validation of labor regulations As in Section 4.3.4, we examine the behavior of the maximum values for each constraint to ensure they remain below their limits. To test this, we set the replacement rate for each contract type and each day to 100%, and increase the probability of employee sickness to 5%, forcing situations that violate the LR requirements to occur. The results are presented in Table 7. We observe that all limits are reached, confirming that the hard constraints function as intended.

Table 7: Maximum observed values for LR constraints in the realization phase.

Constraint	Maximum value observed
Consecutive shift limit	10
Total hours limit	487.5
Weekly hours limit	48
Daily shift limit	1

Extreme inputs First, we analyze the model's behavior when all employees are assumed to be willing to accept new shifts on any given day. In this scenario, we expect nearly all vacant shifts to be filled. Figure 14 shows that this expectation holds for all contract structures with more than 17 employees, while those with 17 employees or fewer still manage to replace at least 99.5% of vacancies. This indicates that the model functions as intended. The remaining vacancies occur in cases where no employees are available, due to existing shifts, sickness, or vacation.

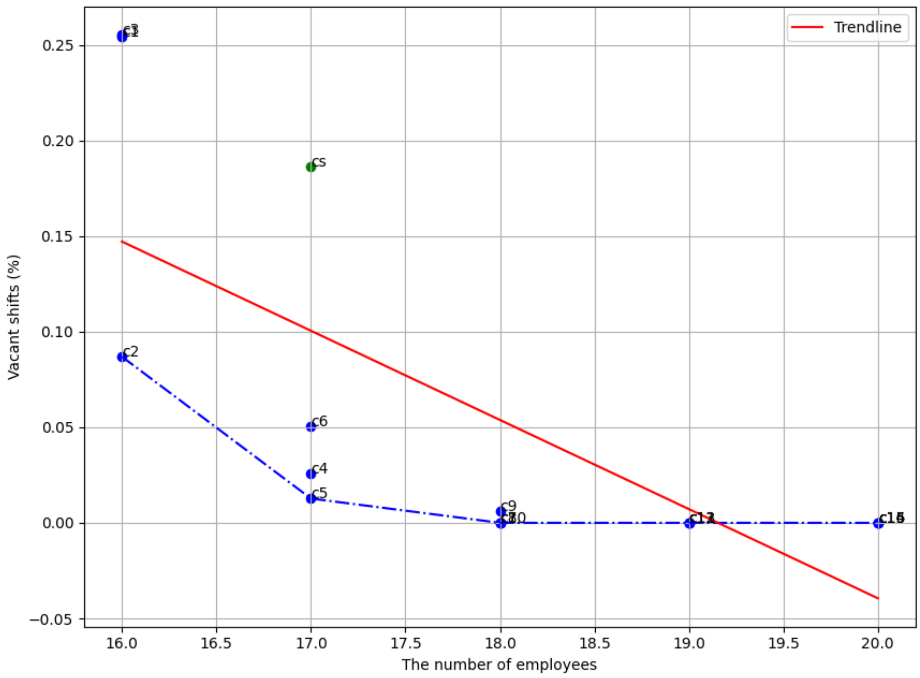

Figure 14: Results where employee willingness to replace shifts is at 100%

Furthermore, we analyze the model's behavior when the sickness rate is set to zero, which should result in no vacancies. The tests confirm that all contract structures behave identically under this condition, no sick leaves occur, and consequently, no replacements are made. This outcome verifies that the model operates correctly in the absence of sickness. The tests were conducted using the set of contract structures defined in Section 4.3.

Consistency of results This analysis is conducted using contract structure 9 (C9). We examine the variance in the utilization of different replacement methods and the sickness rate. Figure 15 presents a box plot for each set of random seeds. The distributions are nearly identical, showing that the model behaves consistently, independent of the chosen seed set. The corresponding figures for the various replacement methods are provided in the appendix (Figures B1, B2, and B3), and they also confirm that the model produces consistent results. Furthermore, this confirms that the chosen number of Monte Carlo iterations (1,000) is sufficient.

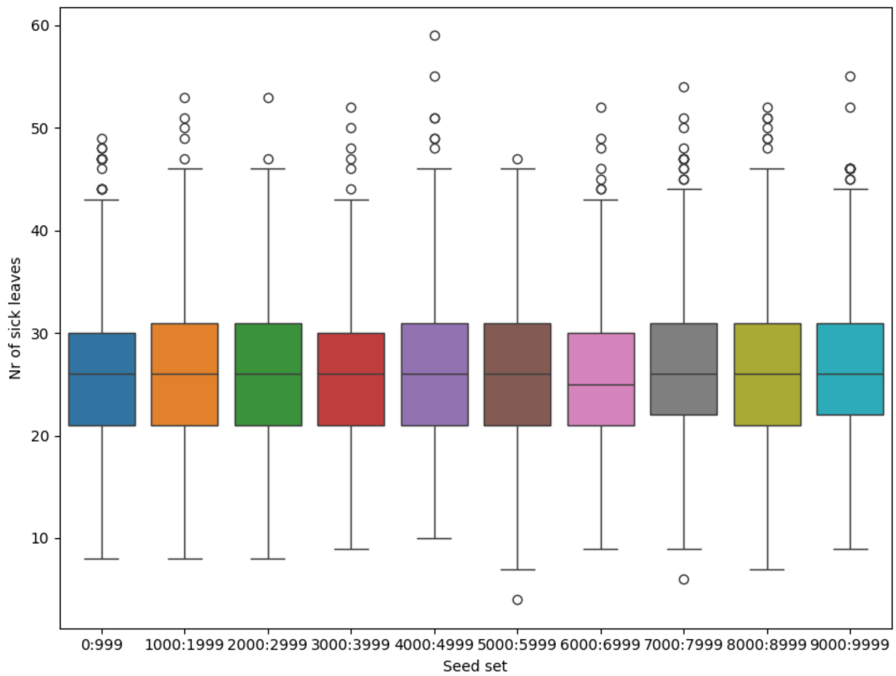

Figure 15: The number of sick leaves using different sets of seeds.

4.4.3 Sensitivity analysis

By studying how small changes to different input parameters affect the output variables, we ensure the model is robust and not sensitive to small changes. We conduct the tests on a single contract structure due to the contract structures' independent nature.

Change in probability an employee falls sick We examine how outputs respond to changes in the probability an employee falls sick p_s using contract structure 9 (c9). An increase in p_s creates more vacancies, we therefore expect the number of sick leaves and vacancies to rise, the absolute number of internal replacements to rise but their share to fall, and external and rental shares to be roughly stable. Table 8 reports the results. As expected, sick leaves increase proportionally to p_s. Vacancies grow faster than the covered share, so the internal share declines as expected, while the rental share trends slightly upwards and the external share remains stable. This pattern is consistent with the assignment policy. As more vacancies remain after internal and external replacements, a larger share is ultimately routed to rental workers, who fill shifts last.

Table 8: KPI values under relative changes to p_s.

Relative change to p_s	−20%	−10%	0%	10%	20%
Number of sick leaves	20.833	23.562	26.065	28.656	31.266
Number of vacancies	7.234	8.445	9.5343	10.548	11.504
Internal replacement share	52.9%	51.2%	50.7%	50.3%	50.2%
External replacement share	6.43%	6.21%	6.55%	6.46%	6.56%
Rental replacement share	5.97%	5.98%	6.19%	6.46%	6.49%

Change in probability an internal employee accepts a new shift We examine how contract structure 9 (c9) responds to uniform changes to the probability that an employee accepts a vacant shift offer $p_{c,w}$. Contract structure c9 is chosen because it includes employees from all groups. Increasing $p_{c,w}$ makes internal employees more willing to accept additional shifts. We therefore expect vacancies to decrease and the internal replacement share to increase, though not one-for-one with the change in $p_{c,w}$. External and rental shares should decline, due to fewer offers. Table 9 reports the results, which align with these expectations. The small variance in the number of sick leaves is simulation noise.

Table 9: KPI values under relative changes to $p_{c,w}$.

Relative change to $p_{c,w}$	−20%	−10%	0%	10%	20%
Number of sick leaves	26.062	26.102	26.065	26.097	26.044
Number of vacancies	11.067	10.177	9.534	8.867	8.121
Internal replacement share	42.9%	47.3%	50.7%	54.3%	58.2%
External replacement share	7.5%	7.1%	6.6%	6.0%	5.4%
Rental replacement share	7.1%	6.6%	6.2%	5.8%	5.3%

Targeted sensitivity for change in probability an employee accepts a new shift In addition to uniform shifts, we change $p_{c,w}$ for Sundays and employees with small contracts ($\leq$ 10 h). Testing $p_{c,w}$ in isolation requires controlling for spillover, studying how the group and other groups are affected.

When testing for small changes in isolation for Sundays, we expect the overall impact to be small, as Sundays constitute only one seventh of data points. Looking only at Sundays, as shown in Table 10, we can note that the behavior is similar to modifying all parameters. Furthermore, in Table C1 we show values for all days, with

$p_{c,w}$ changed only for Sundays. Here we can note that the relative change is small. Furthermore, the change corresponds to the changes observed on Sundays, with both having an absolute difference of 0.4 vacancies, showing the effect on other groups is minimal.

Table 10: KPI values under Sunday-only acceptance change Δ (other days fixed).

Relative change to $p_{c,w}$ (Sundays)	−20%	−10%	0%	10%	20%
Number of sick leaves	3.898	3.872	3.846	3.872	3.894
Number of vacancies	1.869	1.723	1.616	1.531	1.444
Internal replacement (%)	35.2	39.7	43.3	46.2	49.2
External replacement (%)	8.6	8.2	7.7	7.6	7.2
Rental replacement (%)	8.3	7.6	7.1	6.7	6.5

Note. Columns indicate the change applied only to Sunday acceptance; all other weekday–contract cells remain at baseline.

Studying the impact of modifying only small contracts ($\leq 10h$), we first compare replacement rates for individual contract groups. We expect the rate for small contracts to increase, and a slight decrease for other contract types. Furthermore, we expect the overall replacement rate to grow. As contract structure 9 has four employees with contract hours $\leq 10h$ and 15 employees that can perform replacements, we expect the impact to be noticeable. In Table 11, we compare the total number of replacements performed by small contracts and large contracts. It clearly confirms our assumptions. Furthermore, it shows that the impact on replacements done by other contract types is small when considering the absolute number of replacements. Table C2 in the appendix shows the impact on other KPIs. We can note that the overall impact is very small, as we would expect, due to the small change to the internal replacement rate.

Table 11: Internal replacement values under small contract $\leq 10h$-only acceptance change Δ (other contract types fixed).

Relative change to $p_{c,w}$ (Small contracts $\leq 10h$)	−20%	−10%	0%	10%	20%
Number of replacements by small contracts ($\leq 10h$)	3.189	3.501	3.922	4.215	4.705
Number of replacements by large contracts ($> 10h$)	9.616	9.372	9.360	9.326	9.219
Percentage of internal replacements done by small contracts ($\leq 10h$) (%)	33.2	37.4	41.3	45.2	51.0
Internal replacement (%)	49.2	49.6	51.1	52.0	53.5

Note. Columns indicate the change applied only to small contract $\leq 10h$ acceptance; all other weekday–contract cells remain at baseline.

Change in probability an external employee accepts a new shift Continuing with contract structure 9, we expect that a change to the probability an external employee accepts a shift p_{ab} should have no impact on internal replacement share, and decrease rental replacement share slightly, as replacement decisions are ordered. Overall, we expect the number of vacancies to decrease. Table 12 shows that the internal replacement share remains consistent. However, the rental replacement share behaves oddly, seemingly uncorrelated to the external replacement share.

Table 12: KPI values under relative changes to p_{ab}.

Relative change to p_{ab}	−20%	−10%	0%	10%	20%
Number of sick leaves	25.905	25.93	26.348	25.993	26.192
Number of vacancies	9.562	9.391	9.521	9.144	9.112
Internal replacement (%)	51.35%	51.70%	51.36%	51.50%	51.32%
External replacement (%)	5.42%	5.90%	6.56%	7.25%	7.72%
Rental replacement (%)	6.32%	6.19%	5.94%	6.08%	6.17%

Change in probability a rental worker accepts a new shift Utilizing contract structure 9, we expect that a change to the probability a rental worker accepts a shift p_r, should have a small impact on the replacement rate for external employees, as their probability is determined in conjunction with each other, as explained in the methodology. Table 13 shows it behaves in line with our expectations.

Table 13: KPI values under relative changes to p_r.

Relative change to p_{ab}	−20%	−10%	0%	10%	20%
Number of sick leaves	26.107	26.091	26.096	26.209	25.984
Number of vacancies	9.566	9.546	9.345	9.367	9.184
Internal replacement (%)	51.3	51.0	51.4	51.1	51.1
External replacement (%)	6.83	6.69	6.46	6.32	6.30
Rental replacement (%)	5.21	5.70	6.28	6.89	7.25

4.4.4 Summary

The realization phase simulates each day of the planning period, capturing how employees fall sick and create vacant shifts, as well as how these vacancies are replaced. We validated the model performance by confirming that all hard constraints are satisfied, that it behaves correctly under extreme input conditions, and that its outcomes remain consistent across different random sets of seeds. In addition, a comprehensive sensitivity analysis demonstrated that the model responds appropriately to small changes in input parameters and that such changes affect key performance indicators correctly, with the note that rental workers react oddly to changes to the external employee replacement rate.

Based on these results, we show that the realization phase accurately models sickness and replacement processes given the defined input parameters. The resulting outputs provide a sound basis for the evaluation phase.

4.5 Evaluation phase

As the results in the evaluation phase are derived from the realization and planning phases, we do not need to separately validate them. Figure 16 shows the cost and vacancy percentage for each contract structure, ordered by (i) the number of employees and (ii) the vacancy percentage. The dotted line shows costs that arise from undertime and fixed costs. The solid red line depicts fixed costs. We note that substantial undertime represents a far larger cost increase than a slight increase in fixed costs. Moreover, we show that the flexibility can be improved without incurring additional costs.

Furthermore, the issue of overtime among full-time employees, discussed in Section 4.3.1, is negligible. This can be observed, for instance, in contract structure c16, where such undertime occurs.

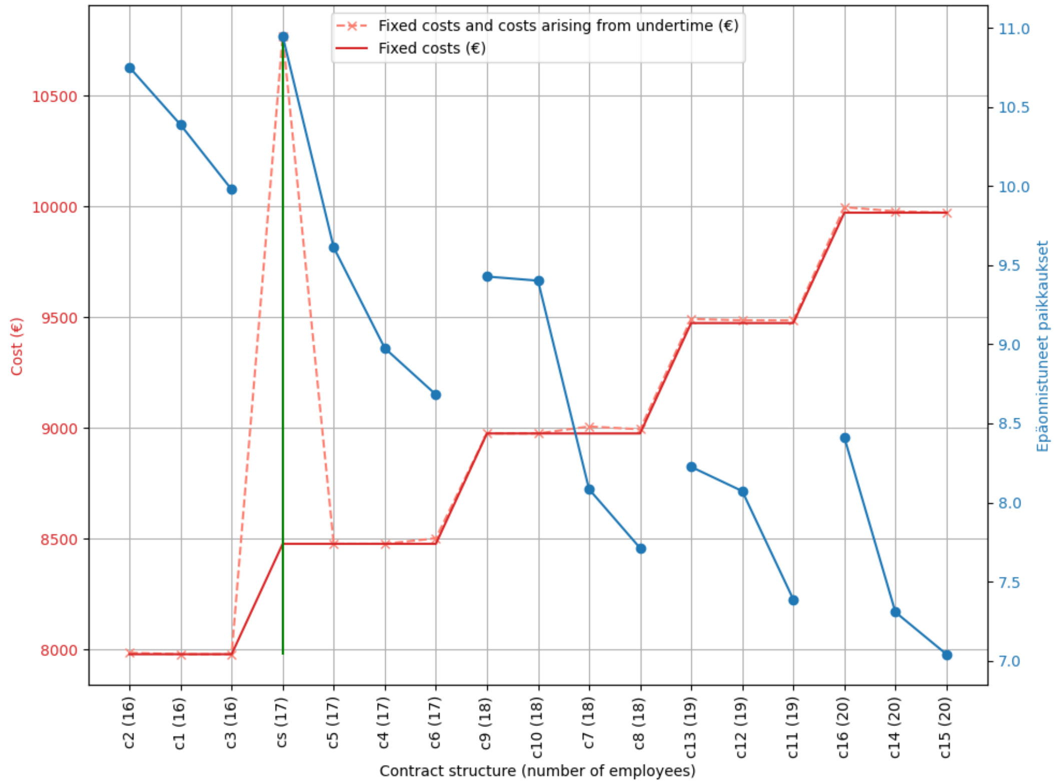

Figure 16: Cost and vacancy percentage for each contract structure. The red dotted line represents total costs, including undertime, while the solid red line shows only the fixed costs. The blue markers indicate the vacancy rate on the secondary axis, and blue lines connect contract structures with the same number of employees. The green line highlights the current contract structure.

To further show the trade-off between minimizing cost and maintaining adequate flexibility, Figure 17 presents the resulting Pareto front. The non-dominated contract structures form a clear frontier, indicating that improvements in vacancy rate beyond this line increase costs. This visualization highlights the effective contract structures, with the Pareto optimal contract structures as: $c_1, c_3, c_6, c_8, c_{11}, c_{14}$, and c_{15}.

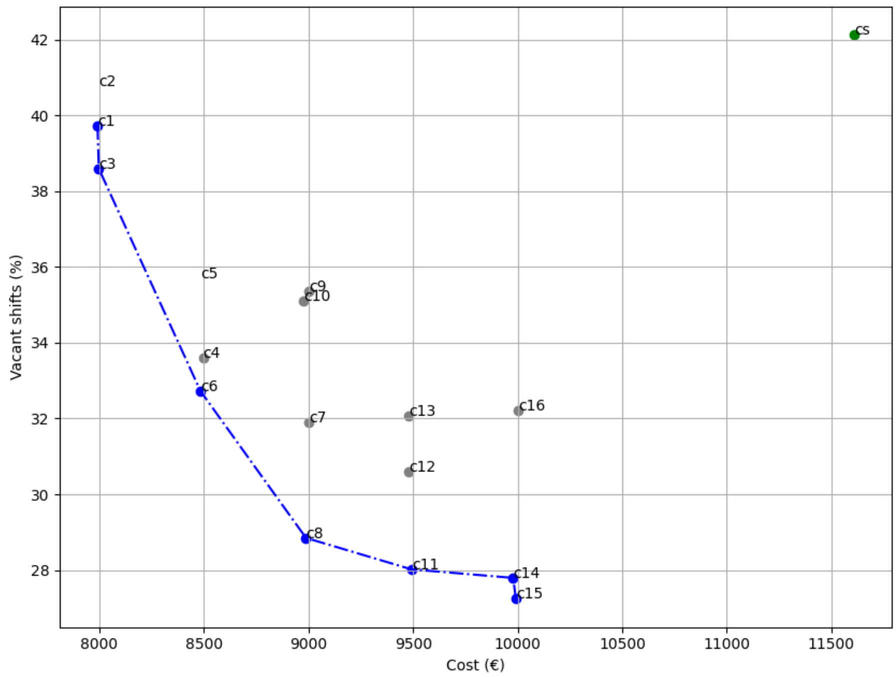

Figure 17: Pareto front showing the trade-off between total cost and vacancy rate across contract structures. Each point represents a contract structure, where the blue line connects Pareto-optimal solutions, indicating the cost of moving between Pareto-optimal contract structures. The grey points denote dominated solutions. The current contract structure is highlighted in green for reference.

From these results, we can derive a structured decision-making process for selecting an appropriate contract configuration:

1. Identify the Pareto-optimal contract structures, ensuring that none include excessive undertime.
2. Determine where improvements in vacancy rate relative to changes in cost are substantial, and filter accordingly.
3. Examine the remaining set of contract structures and select the most suitable one.

In our example, we begin by identifying the Pareto-optimal structures and observing that improvements beyond c_8 yield only marginal reductions in vacancy rate compared to the associated increase in cost. Consequently, contract structures c_{11}, c_{14} and c_{15} are excluded. Since c_1 and c_3 produce nearly identical outcomes, either could be selected. However, we retain c_3 as it offers slightly more flexibility for a negligible difference in cost. This results in a final set of candidate contract structures, c_3, c_6, and c_8, which can then be further evaluated to determine the most suitable contract structure.

Step three requires detailed business insight and should therefore be carried out by the appropriate decision maker. Nevertheless, several observations can be made from the resulting set of Pareto-optimal contract structures. All of these include a substantial number of employees with 15-hour contracts, indicating that this contract type is cost-effective. In contrast, poorly performing contract structures, such as c_2, c_{10}, and c_{16}, contain few or no such contracts. This suggests that 15-hour contracts are cost-effective. However, these contracts are generally unpopular among employees, showing the need to balance contract structure optimality with employee satisfaction.

4.6 Summary

The evaluation phase combines the results of the planning and realization phases. Through this, we can examine how different contract structures perform in terms of cost and flexibility. The analysis confirms that the model captures the expected trade-offs between minimizing costs and maximizing flexibility.

The evaluation results further demonstrate that the model provides clear value to decision-makers, allowing them to identify Pareto-optimal contract structures and assess their relative advantages. The resulting insights show the model's ability to guide strategic decisions regarding contract structure composition. Overall, the evaluation phase confirms the model's usefulness as a tool for workforce planning.

5 Computational results with different parameters

Section 4 validated the model's behavior using an example set of generated contract structures, confirming that it performs consistently and in accordance with the defined constraints. In this section, we extend the analysis by applying the model to a new set of parameters, generating additional sets of contract structures and evaluating their relative performance. This allows us to explore how alternative inputs can provide different insights, valuable to decision-makers in different scenarios.

We adjust parameters defined by the user rather than those empirically estimated from historical data. Such an approach reflects the intended use of the model as a support tool for decision makers, where they can adjust input parameters to evaluate potential scenarios and their implications for cost and coverage.

A detailed analysis can be found in Appendix D. Taken together, the examples demonstrate the framework's usefulness across different use cases. It can be used to examine how vacations affect resourcing needs, to identify the necessary number of employees, to determine how many large contracts the shift schedule can accommodate, and to constrain the set of contracts based on the availability of workers.

These tools provide valuable support for decision-makers in various scenarios:

1. Strategic planning: helping determine the desired direction for developing the contract structure.

2. Short-term resourcing: allowing decision-makers to assess how the contract structure responds to expected demand, for example, in the coming spring, and based on the results, make recruitment decisions.

3. Staff turnover: supporting decisions on whether to recruit a replacement immediately or postpone hiring until demand increases.

4. Structural assessment: identifying whether the current contract structure causes undertime, perhaps the most critical function, given its high cost.

6 Conclusions and discussion

This thesis developed and evaluated a simulation-based model for analyzing contract structures under varying demand and uncertain sickness and replacement processes. The objective was to assess how alternative configurations affect scheduling feasibility, cost, and vacancy outcomes, while explicitly accounting for the constraints imposed by labor regulations (LR) and the differences between contract types. By incorporating LR rules, such as weekly hour limits and required days off, the model ensures compliance with real-world regulations. Distinguishing between contract types allows for more accurate modeling of employee behavior, giving valuable insights into the flexibility of the contract structure beyond the number of employees in it.

We divided the modeling process into four distinct phases: the contract structure generation phase, the planning phase, the realization phase, and the evaluation phase. The phases are designed to mirror real-world decision-making processes, ensuring that the model produces realistic results. In practice, a decision-maker first creates a shift schedule for a given time period (planning phase). This schedule is then affected by unforeseen events such as employee sick leaves (realization phase). Finally, the overall outcome is assessed after the period concludes (evaluation phase). The contract structure generation phase is not a part of the real-world process, but is necessary as an input to evaluate different contract structures, not just the current one.

The contract structure generation phase was implemented using an algorithm based on creating Dirichlet-distributed proportions. The proportions are converted to contract structures and then filtered according to constraints imposed by the decision-maker.

The planning phase was implemented using an algorithmic approach based on a recursive algorithm that searches for the first feasible solution, with all LR rules enforced as hard constraints. The algorithm was designed to ensure fairness by distributing scheduled hours as evenly as possible among employees, both on a weekly and total basis. This guarantees compliance with all LR regulations and ensures employees are treated fairly.

The realization phase was implemented as an agent-based simulation model, representing store employees as autonomous agents that become sick and replace vacancies. External replacements were modeled as pools that perform replacements when internal coverage is insufficient. All agent behavior followed LR regulations, ensuring compliance with real-world constraints. Validation through extreme input and sensitivity analyses confirmed that the model realistically models employee behavior.

The evaluation phase revealed a clear trade-off between cost and vacancy rate, allowing the identification of Pareto-optimal contract structures. These results demonstrate that the model highlights efficient contract structures that balance flexibility with cost. Furthermore, we proposed a practical framework for decision-makers to utilize the model in selecting contract structures. The computational results show the model's value in aiding both long-term strategic planning, such as determining the desired direction for contract structure, and short-term operational decisions, including recruitment and resource adjustments. Together, these examples confirm the model's versatility as a decision-support tool.

The parameters used in the simulation model were empirically estimated. However, certain parameters must remain at the discretion of the decision-maker, such as the selection of available contract types, limits on their availability, and the number of employees taking a winter holiday. Allowing this flexibility ensures that the model can accurately reflect the operational realities of the business, including factors that cannot be fully captured through empirical estimation alone.

Overall, the model serves as a practical decision-support tool for workforce planners, particularly site managers within the S-Group. It enables managers to make more informed decisions regarding workforce composition and planning. Moreover, it has highlighted important factors, for example, that each employee represents a fixed cost and that different contract types behave differently.

Beyond operational improvements, the model contributes to a more data-driven decision-making culture. Historically, many different approaches have been used. The model is an alternative that offers evidence-based insights that encourage more flexible and cost-effective workforce planning.

As the parameters are estimated using data from the entire chain, this generalization is a limitation if substantial variance exists between locations. Furthermore, parameters such as acceptance rate and sickness rate are treated as fixed over the planning period, which is a simplification that does not fully reflect reality. In practice, employee acceptance behavior can depend on factors such as social dynamics and fatigue, which vary over time. Similarly, sickness rates likely follow a cyclical pattern, increasing and decreasing as sicknesses spread through the business.

The agent behavior implemented in the model is also simple compared to real-world conditions, as it does not capture employee preferences. However, given that the model is intended to evaluate potential contract structures rather than individuals, modeling such dynamics would be difficult, as the actual set of employees in the contract structure and their unique preferences remain unknown until they are hired.

Computational limitations include the scheduling process, which identifies the first feasible schedule rather than an optimal one. However, this reflects real-world scheduling practices, which can result in suboptimal but acceptable solutions. Finally, while 1,000 Monte Carlo iterations were shown to be sufficient, increasing the number of iterations could reduce stochastic variability.

The model cannot be directly generalized to other industries, as it is designed specifically for the retail sector and its labor regulations. However, since regulatory constraints exist in any sector, this limitation is not unique to this model but inherent to all workforce planning approaches. A more significant limitation lies in the lack of demand forecasts, as the model currently relies on historical data. This limitation may reduce predictive accuracy and comes with the risk of perpetuating existing problems. Future development should therefore focus on coupling the model with demand forecasting methods to improve accuracy.

The model contributes to bridging the gap between short-term workforce scheduling and long-term strategic workforce planning. Traditional approaches often treat these processes separately, scheduling focuses on maximizing the utilization of employees, while resourcing focuses on staffing levels and cost. Combining employee behavior and LR-based constraints into a single framework enables decision-makers to evaluate how choices at the contract level influence viability, flexibility, and cost.

Theoretically, this work extends the literature on workforce planning, where similar studies have primarily employed optimization-based methods. While optimization approaches are effective for identifying solutions with minimal cost or maximal efficiency under fixed assumptions, they can struggle to capture uncertainty in employee behavior, stochastic sickness dynamics, and labor regulations. In contrast, our approach presented here allows for the exploration of contract structures, including these factors, complementing existing research.

Future work should focus on improving the model's demand forecasting capabilities. In addition, parameter estimation and agent behavior could be improved by including non-fixed sickness and acceptance rates that evolve. External factors could also be modeled in greater detail, rather than being represented simply as a fixed percentage of vacancies not filled by internal employees.

In conclusion, this thesis demonstrates the potential of simulation-based modeling as a practical and theoretically grounded approach to workforce planning. By including empirical parameters, LR-based rules, and employee behavior, the model bridges the gap between scheduling and resourcing. It provides decision-makers with a data-driven tool for evaluating contract structures, balancing cost and flexibility. While further improvements are possible, particularly in forecasting and modeling of agent behavior, the framework is a solid foundation for evidence-based workforce planning.

References

[1] William J. Abernathy et al. “A Three-Stage Manpower Planning and Scheduling Model - A Service-Sector Example”. In: *Operation Research* 21 (1973), pp. 693–711. DOI: https://doi.org/10.1287/opre.21.3.693.

[2] Eric Bonabeau. “Agent-Based Modeling: Methods and Techniques for Simulating Human Systems”. In: *Proceedings of the National Academy of Sciences* 99.suppl_3 (2002), pp. 7280–7287. DOI: 10.1073/pnas.082080899.

[3] George Casella and Roger L. Berger. *Statistical inference*. eng. 2nd ed. Duxbury advanced series. Duxbury, CA: Thomson Learning, 2001. ISBN: 0-534-24312-6.

[4] John J. Forbus and Daniel Berleant. “Using Discrete-Event Simulation to Balance Staff Allocation and Patient Flow between Clinic and Surgery”. In: *Modelling* 4.4 (2023), pp. 567–584. ISSN: 2673-3951. DOI: 10.3390/modelling4040032.

[5] Bela A. Frigyik, Amol Kapila, and Maya R. Gupta. *Introduction to the Dirichlet Distribution and Related Processes*. Tech. rep. UWEETR-2010-0006. Archived from the original on February 19, 2015. University of Washington, Department of Electrical Engineering, 2010. URL: https://vannevar.ece.uw.edu/techsite/papers/documents/UWEETR-2010-0006.pdf.

[6] Pantuso Giovanni. “The Football Team Composition Problem: a Stochastic Programming Approach”. In: *Journal of Quantitative Analysis in Sports* 13 (2017), pp. 113–129. DOI: 10.1515/jqas-2017-0030.

[7] Peter Griffiths et al. “Beyond Ratios - Flexible and Resilient Nurse Staffing Options to Deliver Cost-Effective Hospital Care and Address Staff Shortages: A Simulation and Economic Modelling Study”. In: *International Journal of Nursing Studies* 117 (2021), p. 103901. ISSN: 0020-7489. DOI: https://doi.org/10.1016/j.ijnurstu.2021.103901.

[8] Charles R Harris et al. “Array Programming with NumPy”. en. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

[9] John D Hunter. “Matplotlib: A 2D Graphics Environment”. In: *Computing in Science Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[10] Jean-Daniel Kant, Gérard Ballot, and Olivier Goudet. “WorkSim: An Agent-Based Model of Labor Markets”. In: *Journal of Artificial Societies and Social Simulation* 23 (Jan. 2020). DOI: 10.18564/jasss.4396.

[11] Kristyna Lacinova et al. “ENTIMOS: A Discrete Event Simulation Model for Maximising Efficiency of Infusion Suites in Centres Treating Multiple Sclerosis Patients”. In: *Applied Health Economics and Health Policy* 20.5 (2022). Epub 2022 May 19, pp. 731–742. DOI: 10.1007/s40258-022-00733-0.

[12] J. Lin. “Divergence Measures Based on the Shannon Entropy”. In: *IEEE Transactions on Information Theory* 37.1 (1991), pp. 145–151. DOI: 10.1109/18.61115.

[13] Wes McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[14] Carlos Montoya and Gonzalo Mejia. “Heuristic Algorithm for Workforce Scheduling Problems”. In: *Brazilian Journal of Operations and Production Management* 3 (Jan. 2010). URL: https://bjopm.org.br/bjopm/article/view/BJV3N1_2006_P3.

[15] Christopher Z. Mooney. *Monte Carlo Simulation*. Quantitative Applications in the Social Sciences 116. Thousand Oaks, CA: Sage Publications, 1997. ISBN: 978-0803959439.

[16] Palvelualojen Ammattiliitto (PAM). *Kaupan alan työehtosopimus*. Web page. Accessed: 2025-08-19. 2025. URL: https://www.pam.fi/tes/kaupan-alan-tyoehtosopimus/.

[17] Alessandra Parisio and Colin Neil Jones. “A Two-Stage Stochastic Programming Approach to Employee Scheduling in Retail Outlets with Uncertain Demand”. In: *Omega* 53 (2015), pp. 97–103. DOI: https://doi.org/10.1016/j.omega.2015.01.003.

[18] Plotly Technologies Inc. *Collaborative Data Science*. 2015. URL: https://plotly.com/python.

[19] Prattana Punnakitikashem, Jay Rosenberber, and Deborah Buckley-Behan. “A Stochastic Programming Approach for Integrated Nurse Staffing and Assignment”. In: *IISE Transactions* 45.10 (2013), pp. 1059–1076. DOI: 10.1080/0740817X.2012.763002.

[20] Python Software Foundation. *Python: A Programming Language (Version 3.10)*. 2023. URL: https://www.python.org.

[21] Peer-Olaf Siebers et al. “A Multi-Agent Simulation of Retail Management Practices”. In: *Computing Research Repository* abs/0803.1598 (2008). arXiv: 0803.1598. URL: http://arxiv.org/abs/0803.1598.

[22] Peer-Olaf Siebers et al. “Modelling and simulating retail management practices: a first approach”. In: *Computing Research Repository* abs/1003.3766 (2010). arXiv: 1003.3766. URL: http://arxiv.org/abs/1003.3766.

[23] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. en. In: *Nature* 17.3 (Mar. 2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[24] Michael Waskom. “Seaborn: Statistical Data Visualization”. In: *Journal of Open Source Software* 6.60 (Apr. 2021), p. 3021. DOI: 10.21105/joss.03021.

A Contract structures used in Section 4.

Table A1: Contract structures used in the computational results.

	37.5h	35h	30h	20h	15h	10h	5h
Current contract structure	3	1	6	3	0	2	2
Contract structure 1	3	0	5	4	3	0	1
Contract structure 2	3	0	4	7	1	0	1
Contract structure 3	3	0	3	8	2	0	0
Contract structure 4	3	0	1	11	2	0	0
Contract structure 5	3	0	6	0	6	0	2
Contract structure 6	3	0	2	6	6	0	0
Contract structure 7	3	0	1	10	3	0	1
Contract structure 8	3	0	2	5	8	0	0
Contract structure 9	3	0	6	1	4	0	4
Contract structure 10	3	0	3	8	1	0	3
Contract structure 11	3	0	3	3	8	0	2
Contract structure 12	3	0	1	11	1	0	3
Contract structure 13	3	0	5	1	6	0	4
Contract structure 14	3	0	4	1	8	0	4
Contract structure 15	3	0	0	11	3	0	3
Contract structure 16	3	0	3	8	0	0	6

B Different KPIs utilizing different sets of seeds.

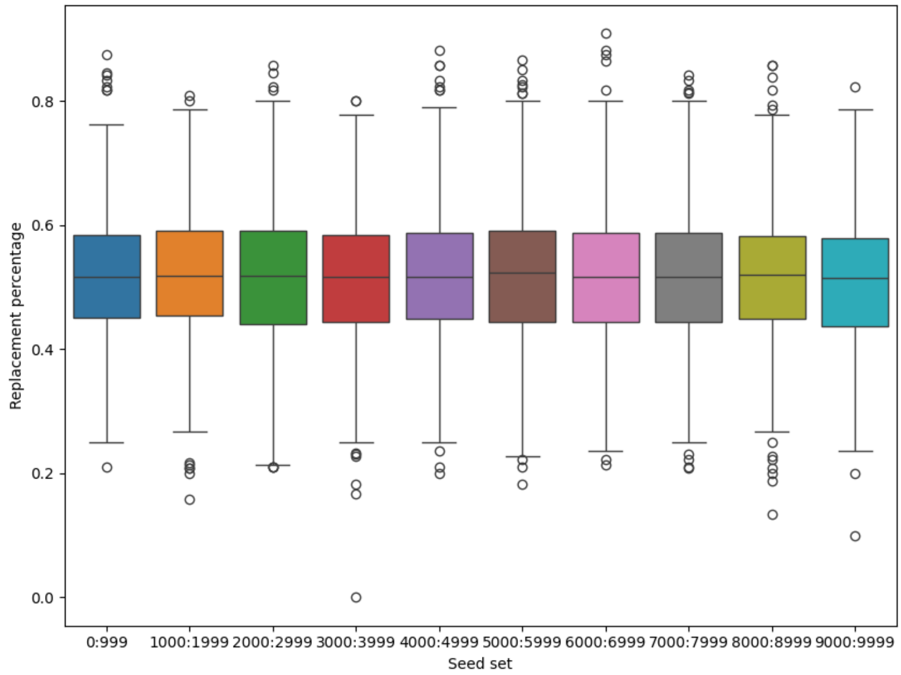

Figure B1: The percentage of replacement done by internal employees using different sets of seeds.

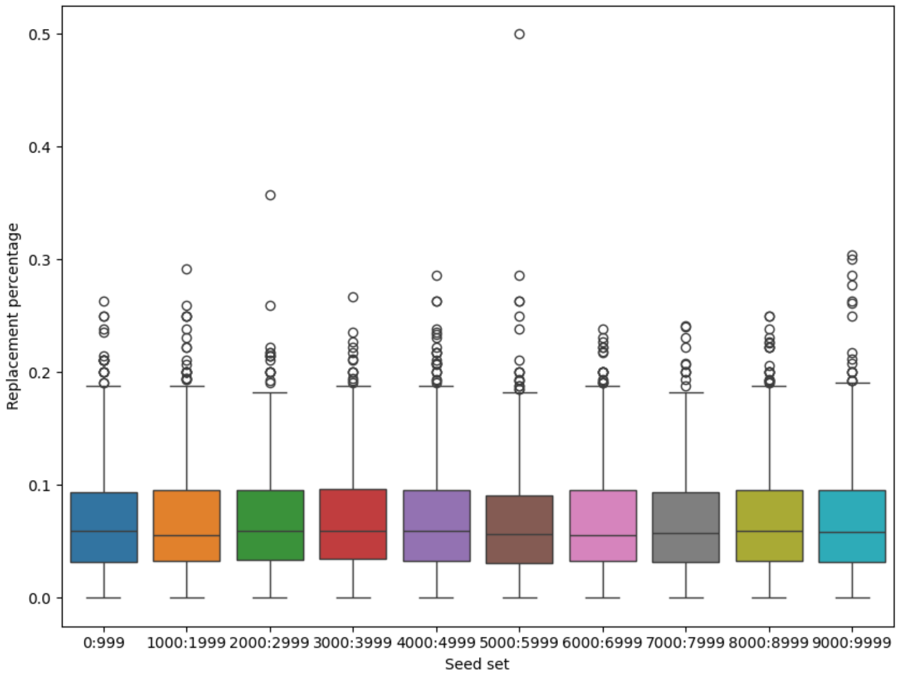

Figure B2: The percentage of replacement done by external employees using different sets of seeds.

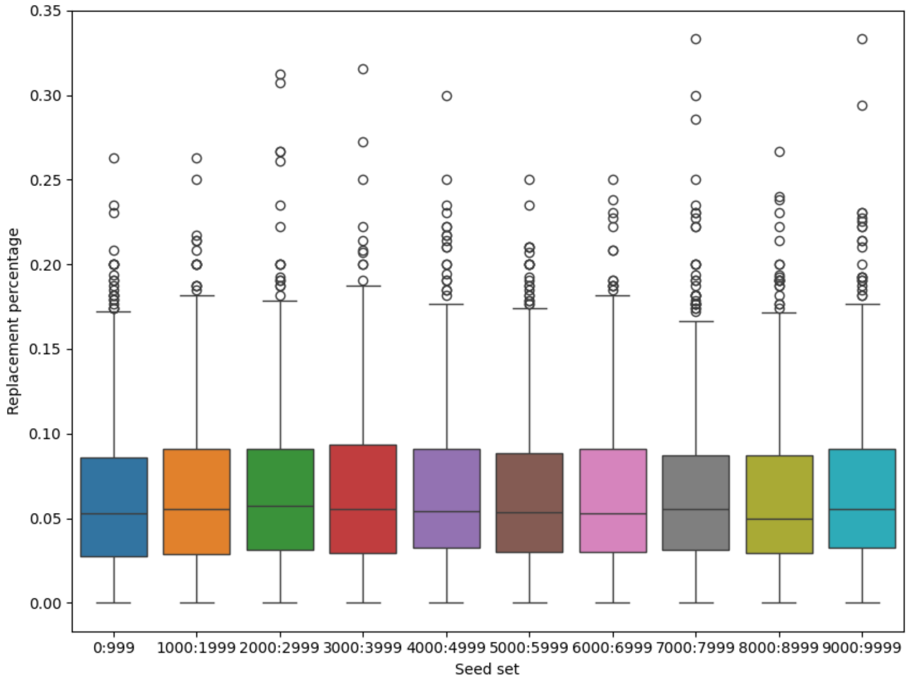

Figure B3: The percentage of replacement done by rental workers using different sets of seeds.

C Additional tables showcasing values from sensitivity testing.

Table C1: All-days KPIs with Sunday-only acceptance change Δ (spillover check)

Relative change to $p_{c,w}$ (Sundays)	−20%	−10%	0%	10%	20%
Number of sick leaves	25.985	25.907	25.830	25.905	25.937
Number of vacancies	9.648	9.458	9.380	9.324	9.257
Internal replacement (%)	49.6	50.4	50.8	51.2	51.6
External replacement (%)	6.8	6.7	6.6	6.6	6.6
Rental replacement (%)	6.5	6.4	6.3	6.2	6.1

Note. Only Sunday acceptance $p_{c,w}$ is changed by all other days remain at baseline. Small, near-constant changes in all-days KPIs indicate negligible spillover from Sunday-only perturbations.

Table C2: KPI values under under small contract $\leq 10h$-only acceptance change Δ (other contract types fixed)

Relative change to $p_{c,w}$ (Small contracts $\leq 10h$)	-20%	-10%	0%	10%	20%
Number of sick leaves	26.155	26.106	26.154	26.236	26.205
Number of vacancies	9.855	9.774	9.771	9.439	9.006
Internal replacement (%)	49.2	49.6	51.1	52.0	53.5
External replacement (%)	6.74	6.47	6.49	6.33	6.18
Rental replacement (%)	6.35	6.49	6.36	6.04	5.98

Note. Columns indicate the change applied only to small contract $\leq 10h$ acceptance; all other weekday–contract cells remain at baseline.

D Calculations of computational results with different parameters

D.1 Example 1: Unconstrained contract generation

We examine the results using a completely unconstrained set of contracts by setting the limits to $L_{min}(h) = 0, \forall h \in [0, 40], L_{max}(h) = \infty, \forall h \in [0, 40], N_{min} = 0$, and $N_{max} = \infty$.We use the same $\delta = 0.05$ and contract types as in Section 4 and generate 16 contract structures to study the model's behavior under these conditions.

As the contract structure generation algorithm is designed to maximize variance, we expect to observe a wide range of possible structures. This is confirmed in Figure D1, where the number of employees varies between 12 and 27. A higher density of number of employees can be seen between 15 and 17 employees. The detailed contract structures for this experiment are provided in the appendix (Table E1).

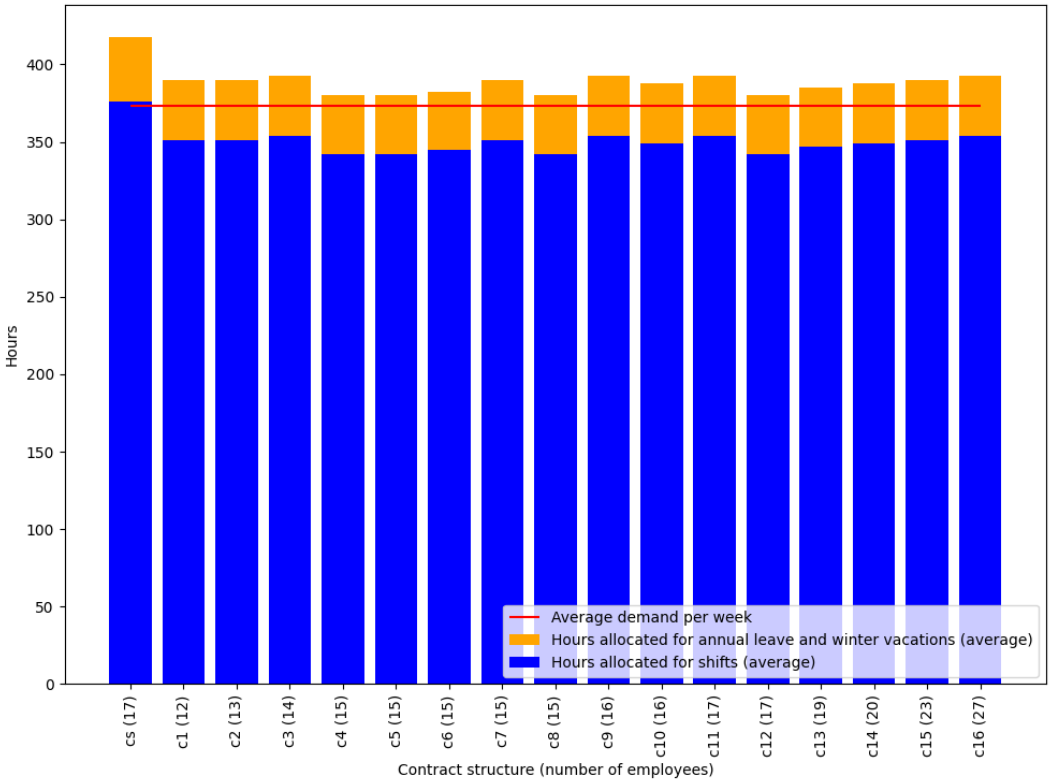

Figure D1: The set of generated contract structures with no limit on number of employees or specific contract types.

Running the generated set of contract structures through the model yields the results shown in Figure D2. We find that contract structures with fewer than 15 employees are unable to produce valid schedules. Furthermore, only two out of the five possible structures with 15 employees pass this phase. Closer inspection shows that 15-employee structures are only feasible when they contain fewer than four full-time employees, likely depending on the number of other large contracts ($\geq 30h$) present in the structure. Notably, structures with up to five full-time employees were able to generate plans without undertime, indicating that the number of full-time employees could potentially be increased.

These results suggest that maintaining a minimum number of 15 employees is essential for schedule feasibility, with 16 employees being preferable, as issues can arise in structures with only 15 employees, depending on the proportion of large contracts ($\geq 30h$). Furthermore, an increase in the number of full-time employees is possible without incurring additional undertime.

We note that the vacancy rate begins to flatten at around 19 employees or more, exhibiting a much slower rate of change beyond this point. However, the limited sample size of contract structures this large makes it impossible to draw exact conclusions.

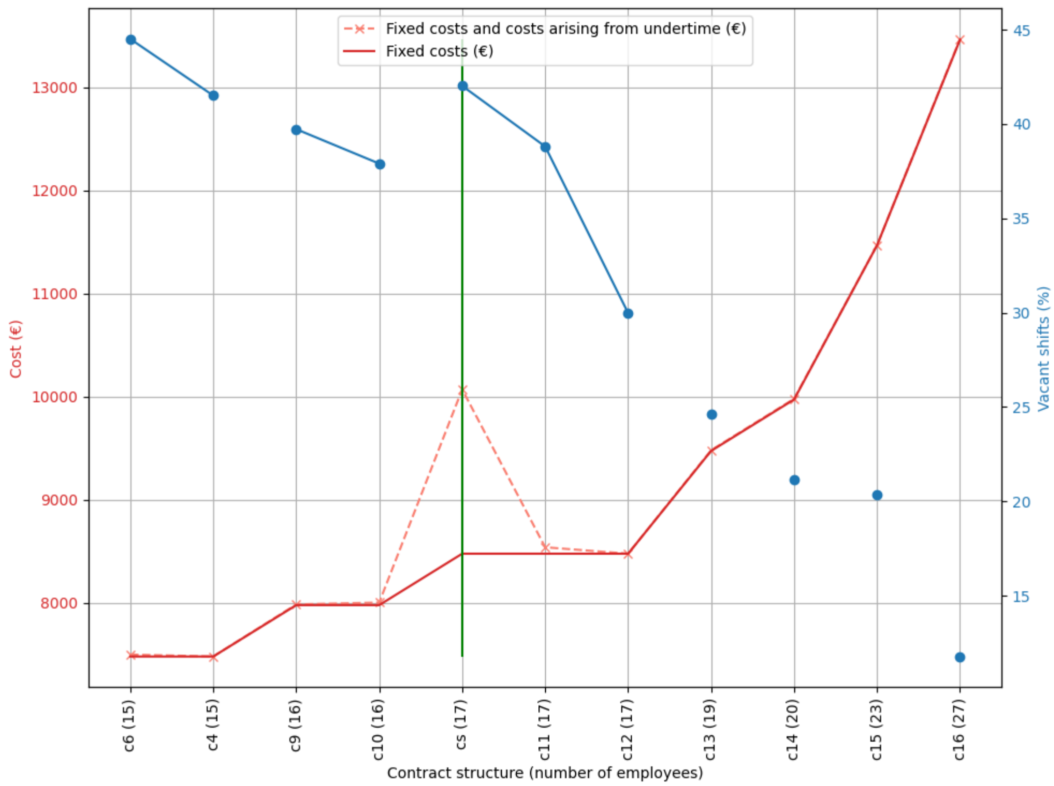

Figure D2: The results with no limit on number of employees or specific contract types.

D.2 Example 2: Constrained mid-sized ($\geq 15h \wedge \leq 25h$) contract types

A common constraint highlighted by decision-makers is the limited interest in mid-sized contracts ($\geq 15h \wedge \leq 25h$) among employees. We impose the restrictions $L_{min}(h) = 0, \forall h \in [15, 25], L_{max}(h) = 1, \forall h \in [15, 25], N_{min} = 0$, and $N_{max} = \infty$. We define the set of available contract types as $T = \{37.5, 30, 20, 15, 10, 5\}$, ensuring an equal number of contract types above and below the constrained range. All other parameters are kept identical to those in Section D.1.

We expect that feasible contract structures will include a relatively large number of smaller contracts. As shown in Figure D3, the generated set of contract structures exhibits high variation, with the total number of employees ranging from 13 to 28. Based on the results from Section D.1, we anticipate that structures with fewer than 15 employees may encounter feasibility issues. The contract structures are shown in detail in Table E2 in the appendix.

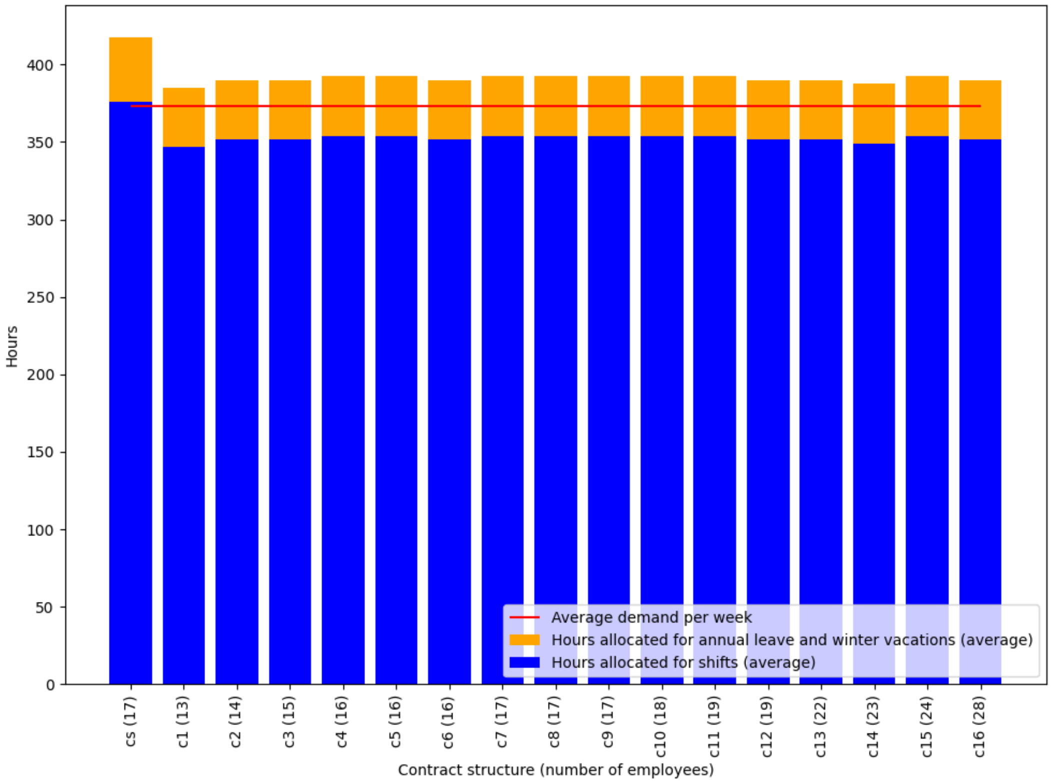

Figure D3: The set of generated contract structures with limits on mid-sized contract types ($\geq 15h \wedge \leq 25h$).

The results obtained from running the generated set of contract structures, shown in Figure D4, indicate that the variance between structures is considerably smaller. This is expected, as all employees capable of replacing vacancies belong to only two contract groups. Since mid-sized contracts are generally the most likely to cover vacancies, the overall vacancy rates remain high.

As anticipated, and consistent with the results of Section D.1, contract structures with fewer than 15 employees are unviable. Furthermore, the structure with 15 employees is also infeasible, as it includes six full-time employees. Notably, undertime issues emerge in structure c13, which likewise contains six full-time employees. Taken together with the findings of Section D.1, these results suggest that an upper limit should be introduced on the number of full-time employees permitted within a contract structure, in this case six.

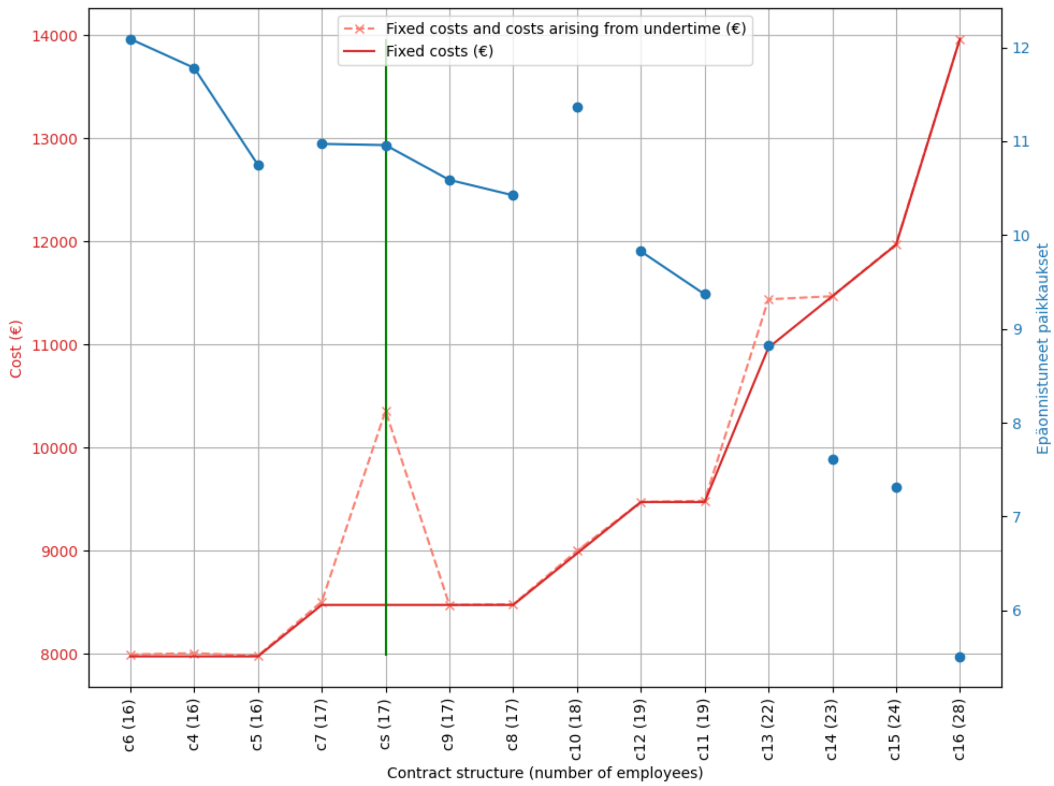

Figure D4: The results with with limits on mid-sized contract types ($\geq 15h \wedge \leq 25h$).

D.3 Example 3: No winter vacations

Solving the model without winter vacations is primarily intended to ensure that it produces relevant results for periods with limited vacation activity, such as autumn, when only regular annual leave occurs.

Since employees take no additional vacations in this scenario, the demand used for contract generation must be updated accordingly. In Section 4.2, the expected total demand is defined as

$$D_{tot} = 0.95 * (D_{avg} + D_{an} + D_{wv})$$

where D_{wv} represents the winter vacations. When winter vacations are excluded, the demand is adjusted to

$$D_{tot} = 0.95 * (D_{avg} + D_{an})$$

Using this updated demand, we generate a new set of contracts employing the same parameters as defined in Section 4.2.

The contract structures obtained, shown in Figure D5, reflect the expected reduction in demand resulting from the lower number of vacations. The current contract structure is clearly over-resourced, exceeding demand by about 50 hours per week on average. Because the number of employees remains the same as in Section 4.4.1, the lower demand results in fewer average working hours per employee. Thus, we expect to observe lower vacancy rates. The detailed contract structures are presented in Table E3.

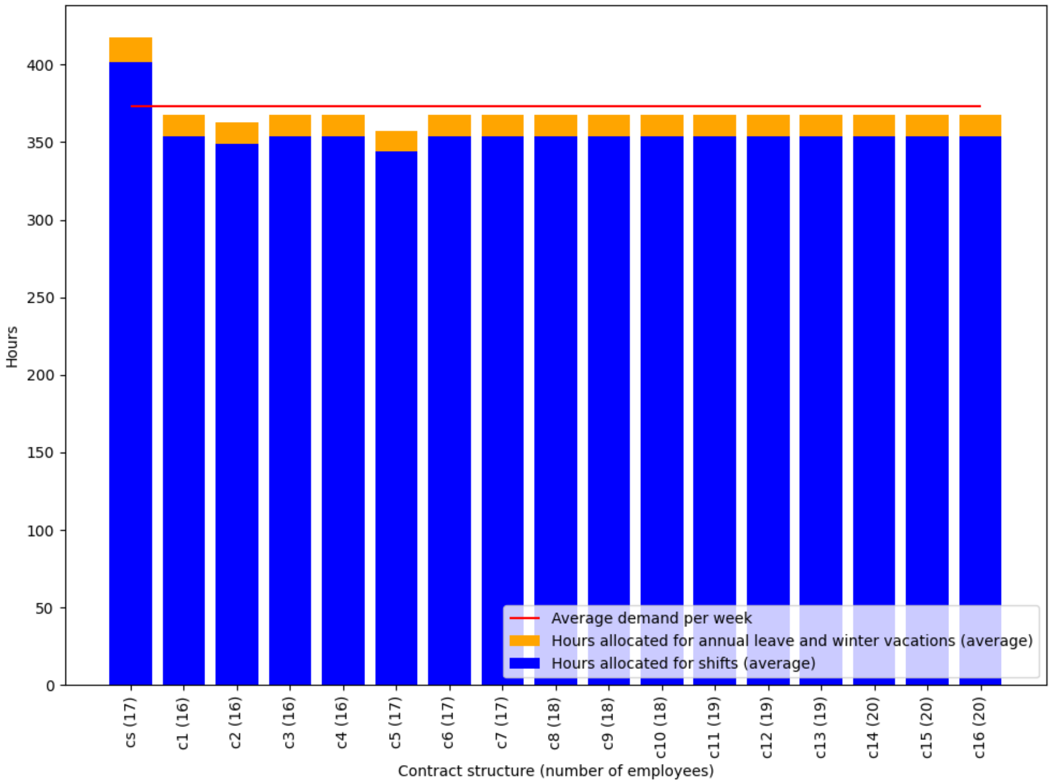

Figure D5: The set of generated contract structures with no winter vacations.

Comparing the results shown in Figure D6 to the example results in Figure 16, we observe that the overall vacancy rate is lower across all personnel sizes, confirming our hypothesis. Additionally, a slight decrease in the vacancy rate is observed for the current contract structure, primarily because employees who were previously on holiday are now available to cover vacant shifts.

Overall, the model behaves as expected, correctly updating the demand to account for the absence of winter vacations. Furthermore, this change introduces no issues in the scheduling algorithm.

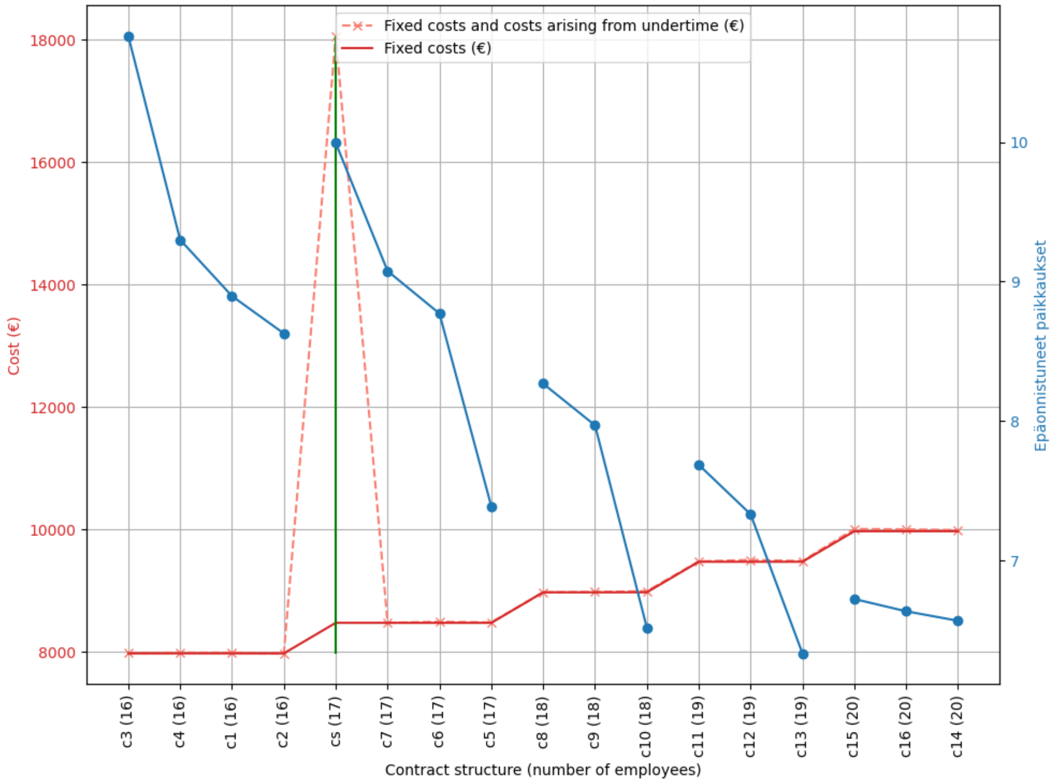

Figure D6: The results with no winter vacations.

D.4 Example 4: Solving for a larger set of contract structures

Finally, we test the model using a large number of contract structures (99) to identify overall trends and determine which types of contract structures are most suitable for the business. Given the extensive number of contract structures, it is deemed unnecessary to restrict the generation process. Instead, we reuse the same set of input parameters as in Section D.1, with the exception that a larger set of contract structures is created.

Due to the large number of generated contract structures, we expect substantial variation in the number of employees, as well as substantial within-group variance. The individual contract structures are not displayed due to the large number of contract structures. For the same reason, a table of all individual contract structures is not included in the appendix.

Out of the 99 generated contract structures, 27 successfully produce a valid schedule. The vacancy rate and personnel numbers for these structures are shown in Figure D7. The within-group variance does not appear much higher than in tests using a smaller number of contract structures, which indicates the effectiveness of the contract structure selection process in the contract structure generation algorithm. Overall, we observe a clear decrease in improvement beyond 21 employees. The smaller number of contract structures in this higher personnel range is due to the limited number of unique combinations possible for such large structures.

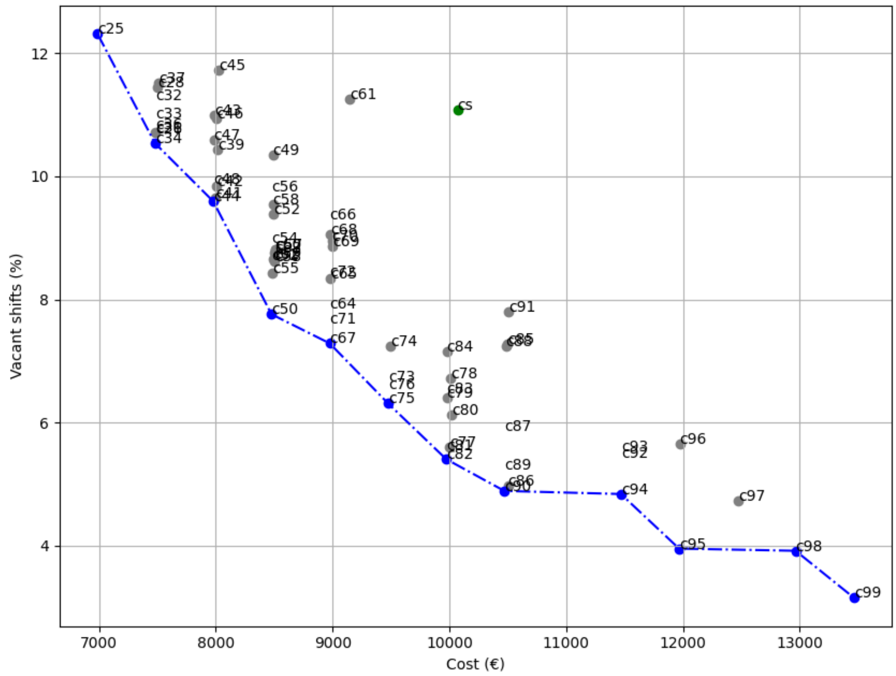

Figure D7: The results with a set of 99 contract structures.

E Contract structures used when testing with different parameters.

Table E1: Contract structures used in Example 1 in Section D.

	37.5h	35h	30h	20h	15h	10h	5h
Current contract structure	3	1	6	3	0	2	2
Contract structure 1	4	0	8	0	0	0	0
Contract structure 2	6	0	4	0	3	0	0
Contract structure 3	1	0	11	1	0	0	1
Contract structure 4	0	0	8	7	0	0	0
Contract structure 5	6	0	0	4	5	0	0
Contract structure 6	3	0	3	9	0	0	0
Contract structure 7	4	0	5	4	0	0	2
Contract structure 8	6	0	1	1	7	0	0
Contract structure 9	3	0	5	4	3	0	1
Contract structure 10	5	0	2	1	8	0	0
Contract structure 11	5	0	0	9	1	0	2
Contract structure 12	0	0	8	1	8	0	0
Contract structure 13	2	0	2	5	10	0	0
Contract structure 14	1	0	0	13	6	0	0
Contract structure 15	0	0	3	12	2	0	6
Contract structure 16	1	0	1	4	14	0	7

Table E2: Contract structures used in Example 2 in Section D.

	37.5h	35h	30h	20h	15h	10h	5h
Current contract structure	3	1	6	3	0	2	2
Contract structure 1	6	0	4	1	0	2	0
Contract structure 2	2	0	9	1	1	1	0
Contract structure 3	6	0	3	1	1	4	0
Contract structure 4	3	0	7	1	1	3	1
Contract structure 5	1	0	10	1	1	1	2
Contract structure 6	4	0	6	1	1	1	3
Contract structure 7	3	0	8	0	1	0	5
Contract structure 8	1	0	9	1	1	5	0
Contract structure 9	1	0	9	1	1	5	0
Contract structure 10	5	0	4	1	1	3	4
Contract structure 11	1	0	8	1	1	8	0
Contract structure 12	2	0	7	0	1	9	0
Contract structure 13	6	0	1	1	1	7	6
Contract structure 14	1	0	6	1	0	15	0
Contract structure 15	1	0	8	0	0	8	7
Contract structure 16	2	0	4	1	1	12	8

Table E3: Contract structures used in Example 3 in Section D

	37.5h	35h	30h	20h	15h	10h	5h
Current contract structure	3	1	6	3	0	2	2
Contract structure 1	3	0	1	9	3	0	0
Contract structure 2	3	0	2	5	6	0	0
Contract structure 3	3	0	6	2	1	0	4
Contract structure 4	3	0	2	8	2	0	1
Contract structure 5	3	0	1	4	9	0	0
Contract structure 6	3	0	1	10	1	0	2
Contract structure 7	3	0	4	3	4	0	3
Contract structure 8	3	0	4	2	5	0	4
Contract structure 9	3	0	1	9	2	0	3
Contract structure 10	3	0	1	3	11	0	0
Contract structure 11	3	0	4	1	6	0	5
Contract structure 12	3	0	1	8	3	0	4
Contract structure 13	3	0	2	1	11	0	2
Contract structure 14	3	0	3	1	8	0	5
Contract structure 15	3	0	0	10	2	0	5
Contract structure 16	3	0	2	4	6	0	5