Heuristics for days-off scheduling of heterogeneous workforce

Tommi Summanen

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 30.11.2021

Thesis supervisor:

Prof. Fabricio Oliveira

Thesis advisors:

M.Sc. Laur Pulliainen

D.Sc. Paul Saikko

Aalto University
School of Science

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author: Tommi Summanen

Title: Heuristics for days-off scheduling of heterogeneous workforce

Date: 30.11.2021 Language: English Number of pages: 6+38

Department of Mathematics and Systems Analysis

Professorship: Mathematics and Operations Research

Supervisor: Prof. Fabricio Oliveira

Advisors: M.Sc. Laur Pulliainen, D.Sc. Paul Saikko

Flexible working life and competition between companies characterize modern working environments. To address this, employers use automatic workforce scheduling to reduce manual planning work and optimize schedules on some aspects, such as costs or employee satisfaction. In this thesis, the problem of allocating rest and workdays of employees is modelled as a mixed-integer nonlinear optimization problem that features a workload forecast consisting of multiple task types, varying employee skills on different task types, and employee-specific contracts. The goal of the thesis is to find a performant method for producing high-quality solutions for the model. The studied method is solving a relaxed problem instance and constructing an integer solution from the real number solution with two heuristics. Solution times, solution quality, and feasibility of the solution obtained were analyzed using a data set of 60 realistic scenarios. The heuristics provided high-quality solutions in terms of feasibility and objective function values. Regarding solution times, obtaining the real number solution was recognized as a performance bottleneck.

Keywords: workforce optimization, days-off scheduling, heuristic, heterogeneous workforce

AALTO-YLIOPISTO
PERUSTIETEIDEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Tommi Summanen

Työn nimi: Heuristiikkoja heterogeenisen työvoiman työpäivien aikataulutukseen

Päivämäärä: 30.11.2021 Kieli: Englanti Sivumäärä: 6+38

Matematiikan ja systeemitieteiden laitos

Professuuri: Matematiikka ja operaatiotutkimus

Työn valvoja: Prof. Fabricio Oliveira

Työn ohjaajat: FM Laur Pulliainen, FT Paul Saikko

Nykyaikaisille työympäristöille on ominaista joustavuus ja yritysten välinen kilpailu. Työnantajat käyttävät automaattista aikatauluttamista vähentääkseen manuaalista suunnittelutyötä ja optimoidakseen aikatauluja mm. kustannusten ja työntekijätyytyväisyyden osalta. Tässä diplomityössä työntekijöiden lepo- ja työpäivien allokoimisongelmaa mallinnetaan epälineaarisena kokonaislukuoptimointiongelmana, jossa työntekijöiden taidot vaihtelevat eri tehtävissä ja työntekijöillä on yksilölliset työsopimukset. Tutkielman tavoitteena on löytää suorituskykyinen menetelmä laadukkaan ratkaisun tuottamiseksi malliin. Malli ratkaistaan laskemalla ongelmainstanssille reaalilukuratkaisu, jossa kokonaislukurajoitukset jätetään huomiotta. Reaalilukuratkaisusta muodostetaan kokonaislukuratkaisu kahdella heuristiikalla. Algoritmien testaamiseen käytettiin 60 realistista testidatainstanssia. Ratkaisuajat, ratkaisun laatu ja saadun ratkaisun rajoitustenmukaisuus analysoitiin. Heuristiikat tuottivat rajoitusten noudattamisen ja tavoitefunktion arvojen kannalta laadukkaita ratkaisuja. Ratkaisuaikojen osalta reaalilukuratkaisun saaminen todettiin suorituskyvyn pullonkaulaksi.

Avainsanat: työaikataulujen optimointi, heuristiikka, heterogeeninen työvoima

Preface

In a sense, this master's thesis was a team effort. I thank every teacher from the kindergarten, comprehensive school, high school, Aalto University, or other contexts who have influenced my growth intellectually or as a person and made me possible to get this far. Sadly few of them will ever see these acknowledgments, but I believe they know anyway, someway.

I thank Laur Pulliainen, Paul Saikko, and Alexander Bor, who was my advisor part of the time, for always having time to help me with my thesis. Markus Mäkynen made sure that I was able to concentrate on the thesis. Fabricio Oliveira supervised, gave helpful feedback, and made sure that all practicalities went smoothly. Special thanks to Henrik Sjöström for his jokes and organizing a topic for my thesis. In addition, my friends, Fyysikkokilta, Fyysikkospeksi, and the most awesome colleagues in the Business Support and Testing team have made past years fun and worthwhile.

Espoo, 30.11.2021

Tommi Summanen

Contents

1 Introduction

In modern work environments, automated workforce scheduling is an increasingly important topic, driven by many economic and societal trends. Competition drives companies to cost reductions by optimizing processes, investments in the welfare and satisfaction of employees create demand for tools that support these goals, and meeting complex restrictions of agreements is easier with automatic procedures than by manual planning. The service sector has grown in importance in past decades. A big proportion of costs in the sector are labor costs, making it widely studied in personnel scheduling literature.

This study focuses on investigating performant solution methods for a particular workforce scheduling model with multiple objectives, multiple task types and heterogenous employees. The model covers the decision of allocating rest and workdays of a fixed pool of employees over a given time period. Related aspects such as hiring and firing employees, or estimating the workload, are not included in the scope. The model is not targeted at any specific industry, but it is rather intended for general use. Instead of calculating the optimal solution, the goal is to obtain a feasible high-quality schedule with reasonable computational resources.

The studied method is constructing an integer solution with a heuristic from a real number solution of a problem instance, where integer constraints are omitted. Two heuristics are presented and compared. Solution times, objective function values and feasibility of solution are tested on a realistic data set of 60 instances with varying parameters and problem size.

This study is structured as follows. In chapter 2 literature related to personnel scheduling is reviewed. Application areas, modelling, and solution methods of this versatile problem field are discussed. Chapter 3 introduces a personnel scheduling model that is at the center of this thesis. Assumptions of the model, objective functions, soft constraints, and hard constraints are presented. Two heuristic methods for solving the model are proposed in section 4. Computational results for heuristics on a realistic data set are reported in chapter 5. Data set is presented, and solution times, solution quality, and feasibility of solutions are discussed. Chapter 6 concludes the study by summarizing the main findings and suggesting directions for future research.

2 Background

In this chapter, literature related to personnel scheduling as an optimization problem is reviewed. Different aspects related to the problem such as application areas, ways to classify model, estimation of demand, planning periods, employee characteristics, typical constraints, typical objectives as well as solution methods are discussed. The review paves way for subsequent chapters where a particular staff scheduling model is introduced and results from the model using a realistic data set are assessed.

2.1 Application areas

Van den Bergh et al. [1] categorize personnel scheduling problems based on the application area. Nurse scheduling is the most studied problem domain in their material. The work of nurses is often manual and labor-intensive, and meeting staff level, training, and seniority requirements is important. Burke et al. [2] present an overview of papers related to nurse scheduling.

Other application areas that have received a lot of interest according to Van den Bergh et al. [1] are call centers, manufacturing, transportation, and retail. Work in call centers is characterized by varying demand, handling several types of calls whose handling requires different competencies, varying random length of calls, and customer's tendency to drop calls after a certain amount of waiting. Defraeye et al. [3] offer an overview of staffing and scheduling under nonstationary demand for service and the majority of papers in the review are inspired by work in call centers.

Van den Bergh et al. [1] note that there are more papers in services than in manufacturing in general. Personnel scheduling in services is characterized by a varying demand that must be met immediately, while in manufacturing demand is more predictable and other means than the number of personnel such as storages can be used to react to changing demand. The need for optimizing personnel scheduling in production environments has however increased because of quick-response and personalized manufacturing [4].

Personnel scheduling in transportation has its specifics: for airlines and railways allocating crews to operate vehicles is typical rather than allocating individual employees [5, 6]. Typical for workforce scheduling in retail is competition between companies which drives them to balance between strict cost management and good service level [7].

2.2 Classification

Baker [8] proposed a classification of workforce scheduling problems based on the decision type that is achieved with the model. Three distinct problem types are recognized: shift assignment, off-days scheduling, and tour scheduling problems. In shift assignment problems, workforce is allocated between different tasks inside one-day planning horizons. In the most simple form of this problem, shifts do not overlap but, in more complex scenarios, overlapping is allowed. In off-days scheduling problems, on the other hand, rest and workdays of workers are allocated instead of assigning them to specific tasks. The need for such decisions arises when the length of the operating week of the employer and the length of the workweek of the employee are different. This happens for example if a shop is open seven days a week but employees work only five days during the week. The third category is tour scheduling problems. These combine shift assignment and off-days decisions into one single problem, being hence the most complex type of personnel scheduling problems. Van den Bergh et al. [1] point out in their literature review that the trend is towards tour scheduling problems as personnel preferences are taken more into account and, in general, the work environment is becoming more flexible.

Another important division is between cyclical and acyclical personnel scheduling problems. In a cyclical problem, each employee performs the same sequence of task assignments, with the only difference between employees being different starting times. This may lead to structure in the model that can be exploited to build efficient solution algorithms.

2.3 Workload estimation

Regardless of whether the problem at hand is to schedule shifts, working days, or both, the core of the problem is to match the schedule to an estimation of the required amount of work considering problem-specific criteria and limitations. According to Ernst et al. [9], much of the rostering literature assume that this demand is either given or can be obtained without difficulty. Separating personnel scheduling and demand forecasting reduces the complexity of models and many times this separation arises naturally. In some scenarios, schedule and demand are interconnected such as when understaffing leads to backlogged demand that increases workload later but, according to Ernst et al. [9], these effects are usually small.

Ernst et al. [9] categorize demand into three different types: task-based demand, flexible demand, and shift-based demand. If demand consists of a list of tasks that are known in advance, such as in transportation applications where timetables of vehicles determine the need for staff, then the demand is task-based. Flexible demand refers to situations where service requests arise randomly and also the time it takes to complete a service is possibly random. Call centers and shops show this kind of behavior. Context knowledge is needed to convert service forecast to the required number of workers or working hours. Shift-based demand arises in situations where the number of required staff is known beforehand. In nurse scheduling, staff levels are often based on regulations and the number of patients leading to shift-based demand.

Brusco and Johns [10] note that in the literature, the mean, shape, and amplitude of demand, defined as the maximum difference of labor requirements across the week in any given hour of the day, are usually varied when algorithms are tested with synthetic data. They also show that smoothness, defined as the period to period fluctuation of the demand, affects the performance of heuristic solutions.

2.4 Modelling

The planning period is the time frame in which scheduling is performed. A variety of planning periods are seen in the literature. Franz and Miller [11] study assigning medical residents to training rotations and clinic experiences and use a one-year planning period. Bailey and Field [12] introduce a flexible scheduling method they call flexshift that gives employees a possibility to select shifts that are most suitable for them and use a one-week planning period to evaluate the performance of the model. Often models are formulated such that changing the planning period is straightforward.

Early literature frequently assumes homogenous employees. That is, every employee has the same set of skills, same productivity, same possible working hours allowed by the contract, etc. – employees are similar in all dimensions that are relevant for the model. A more realistic assumption of a heterogeneous workforce allows variation in some or all of the relevant dimensions.

Depending on the complexity of the work, dividing it to separate task types and requiring workers that are skilled for a specified task before assigning it to them might be relevant. Rong [13] presents monthly tour scheduling models for service industries where an employee can have any combination of multiple skills and the cost of assigning an employee to work varies being the highest for multiskilled employees. Henao [14] optimizes training of employees in multiple skills and concludes that relatively little training is needed to obtain optimal costs when compared to all possible employee-skill combinations that could be achieved with training. In addition, some models take into account how skillful an employee is in a task instead of just acknowledging whether the employee can perform that task or not. Firat et al. [15] deal with assigning jobs to technicians in a setting where a job is divided into skill domains, employees have skill levels associated with each skill domain and every job has a minimum skill requirement. The idea of varying skill levels is closely related to modelling productivity of employees: Walter and Zimmermann [16] give employees coefficients that describe how long it takes for a worker to perform one hour on budgeted work in their paper on a multi-project staffing problem.

Some authors include the experience of the workforce in their models. Wang et al. [17] include improvement of employee skills over time in a model where short-term productivity and long-term human resource development are balanced. Topaloglu [18] studies staff scheduling of healthcare residents in a highly constrained setting where more senior residents are favored when less desired shifts such as Friday shifts are assigned.

Authors differ whether they study full-time, part-time workers, or both. According to Van den Bergh et al. [1], using full-time workers is the most studied setting. Shoewalter and Mabert [19, 20] find that using part-time workers in the employee pool allows a better match of allocated work to the workload. Sometimes use of an external workforce is allowed to cover unfulfilled workload such as in the model by Valls et al. [21] for service centers.

A natural objective for personnel scheduling is to build a minimal roster where the forecasted workload in the planning period is met with as small a workforce as possible. Dantzig [22] in one of the first papers on the workforce scheduling minimizes the total amount of employees starting a shift during the planning period. Di Caspero et al. [23] minimize three objectives in their discussion on the minimum shift scheduling problem: deviations both up and down from the ideal amount of personnel, as well as the number of distinct shift types in the roster. Pastor and Olivella [24] incorporate working time accounts of employees to the staff scheduling problem formulation of a retail clothing chain to allow short-term flexibility in scheduling while maintaining long-term balancing of work hours.

Van den Bergh et al. [1] note that minimizing costs is more flexible and hence, a more popular objective in the literature than minimizing the amount of workforce. Ingels and Maenhout [25] study the optimal trade-off between staff size and the use of overtime by including understaffing, salary, and overtime costs in the objective function. The authors also run simulations with varying staff sizes and overtime budgets. Fowler, Wirojanagud and Gel [26] divide employees to skill groups and allow individual differences concerning "general cognitive ability", and, in the objective function, individual training, hiring, firing, and missed manufacturing costs that depend on the skillset and the cognitive ability level are included. Joubert and Conradie [27] include costs of hiring casual workers in their objective function and they also allow for cost differences between days of the week by compensating work on Saturdays better than everyday work. Nasir and Kuo [28] include travelling, vehicle maintenance, and workforce costs in a model for allocating home healthcare workers where multi-vehicle routing and staff scheduling problems are integrated into one problem.

The interest of authors is not limited to minimizing working hours or costs. In the model by Al-Yakoob and Sherali [29] employees submit personal preferences for specific shifts, work locations, and days-off, and in the objective function differences between wishes and schedule are minimized to improve employee satisfaction. Alsheddy and Tsang [30] form an objective function from two components in their empowerment scheduling approach of field workforce: in the first objective function, the sum of priorities of job assignments is maximized, and in the second objective function, the match between schedule and employee wishes is rewarded. Shahnazari-Shahrezaei et al. [31] optimize the match between employee skills and skill requirements of the workload by penalizing assignment of overqualified workers, e.g., senior workers to junior-level tasks in their multi-skill staff scheduling problem. Wang et al. [17] minimize processing times in precast production in a setting where employees become more efficient over time.

Scheduling enough rest in form of days off and breaks is important in staff scheduling and often achieved with constraints. Other explicit health and welfare-related goals have been included in models. Finco et al. [32] improve ergonomics in the assembly line by using a smoothing index that describes equality of the distribution of work as an objective function. Ayough et al. [33] minimize boredom experienced by the workforce, which they define as exposure to similar tasks. Xu and Hall [34] give a broad overview of how minimizing fatigue has been incorporated in personnel scheduling models.

In highly constrained settings, obtaining feasible solutions is challenging and can be an objective itself. Topaloglu [18] minimizes deviations from soft constraints in a highly constrained medical resident scheduling problem and acknowledges that constraints may be conflicting sometimes. Similarly, Lü and Hao [35] form an objective function from deviations from soft constraints in their nurse scheduling problem.

In staff scheduling problems, it is typical that multiple separate objectives are pursued simultaneously. The planner in those cases must give objective functions weights that reflect their relative importance and combine them to a single objective function.

Personnel scheduling problems are often highly constrained, containing both soft and hard constraints. The solution for the model is not allowed to break hard constraints. For soft constraints, violating them is allowed but undesired and the total violation is minimized.

Coverage constraints enforce that shift assignments match the predefined workload. Depending on the modelling details, it may be expressed as a requirement for working hours during a time period, the number of people starting a shift, or the total number of people working during the day. Van den Bergh et al. [1] note that the hard coverage constraint is the key characteristic of personnel scheduling problems appearing in 75% of the papers they reviewed, and coverage constraints are also common as soft constraints. Both soft and hard coverage constraints can simultaneously exist in the model: in the study by Bard [36], the right input data for optimizing the number of permanent staff at a mail processing and distribution center is investigated and the number of full-time and part-time employees is included as a hard constraint as well as an objective function. The attitude towards under and overstaffing varies. Elshafei and Alfares [37] require that the number of workers each day matches exactly the demand of that day in their days-off scheduling problem with sequence-dependent labor costs, while Hochbaum and Levin [38] allow for overstaffing in their study of the complexity of algorithms for multi-shift scheduling problem. Gärtner et al. [39] allow for both under and overstaffing and propose a greedy search based algorithm for shift and break scheduling. Van den Bergh et al. [1] note that allowing only understaffing is the rarest situation and refers to the study by White et al. [40] of scheduling clinical training staff as the only example of such setting among papers they reviewed. In the model by White et al. [40], ideally five persons are allocated to work at each night shift, but due to chronic understaffing, often fewer people are allocated.

A variety of worktime-related constraints are seen in personnel scheduling models. These constraints arise from legislation, collective agreements, employee-specific contracts, and organizational policies. Van den Bergh et al. [1] list the following types of time related constraints that have been used either as soft or hard constraint among papers they reviewed: maximum/minimum number of assignments, maximum/minimum number of assignments to a shift type, maximum/minimum number of consecutive days, maximum/minimum number of consecutive days off, maximum/minimum number of working hours, maximum number of overtime, days and shifts for which employees have announced beforehand that they will be on or off, time between assignments, restrictions on shift patterns, restrictions on consecutive shifts, restrictions on shift type sequences, free days after night shifts, maximum number of weekends with work in a given number of weeks, number of complete (and extended) weekends, maximum number of extended weekends, number of identical weekends, acceptable time windows and deadlines, resource-availability related constraints and ratio of different worker groups related constraints. The interested reader may refer to Van den Bergh et al. [1] for a detailed discussion about time-related constraints in personnel scheduling literature.

Many other constraint types exist in the literature. Some groups of workers may be given privileges over other groups. Topaloglu [18] favors residents who have worked for a longer time when assigning free weekends and Friday shifts in a medical staff scheduling problem. In multiskilled settings, skill-related constraints are typical. Firat et al. [15] study branch-and-price algorithm in hierarchical skill setting where workers on a higher ladder of skill hierarchy can perform tasks where a lower level of skills are required but not vice versa. Limited resources that are needed to perform work may also be modelled with constraints. Maghzi et al. [41] minimize the total amount of time spent by patients in hospital wards and model limited capacity of operating rooms by placing patients in a waiting queue. The recent COVID-19 pandemic has introduced new challenges to personnel scheduling as also health safety of workers must be addressed. Organizations have to consider issues such as a maximum number of onsite workers to maintain social distancing [42], or dividing workers into mutually exclusive groups to minimize the risk of contagion [43], and include these as constraints to staff scheduling models.

2.5 Solution methods

In a survey of the tour scheduling literature by Alfares [44], it is noted that large size and integer nature are typical for personnel scheduling problems. Therefore, computational resources frequently become an issue, especially when the exact solution is calculated. This, together with the versatility of staff scheduling models, has led authors to propose a wide variety of inexact methods to approximate the optimal solution with reasonable computational resources. Alfares [44] proposes a categorization of papers based on solution methods and notes that categories can be grouped to optimal and heuristics approaches. Optimal approaches include manual solution, integer programming, and implicit modeling. Heuristic approaches include decomposition, goal programming, working set generation, LP-based methods, construction and improvement, metaheuristics, and other methods. While there exist exact decomposition methods, in this categorization they belong under integer programming, and only inexact methods are grouped under decomposition methods. Next, this categorization is discussed. A separate section is dedicated to LP-based solution approaches.

Manual solution, integer programming, and implicit modeling are exact solution methods. Manual exact solutions are possible only in very simple problem settings. Alfares [45] proposes the optimal workforce schedule obtained by manual inspection for the main security gate of a large oil company with a varying number of open lanes during the day, homogenous workforce, and cyclical schedule. Using well-known exact integer programming methods such as branch-and-price is often challenging due to the long computing time requirement. Providing an initial guess for the solution and utilizing decomposition approaches when the structure of the problem allows are known ways to address challenges with computation times. Yaoyuenyong and Nanthavanij [46] calculate the upper bound for solutions with a heuristic method to help the convergence of an exact branch-and-bound based method in an energy-based workforce scheduling problem. Zhu and Sherali [47] apply Bender's decomposition to a workforce planning problem with multiple units and demand uncertainty. In general, combining many methods in the solution procedure is typical for staff scheduling problems. In implicit modeling, the problem is reformulated to reduce the number of decision variables: Bechtold and Jacobs [48] present both implicit and explicit problem formulation of shift scheduling in a hypothetical organization with flexible break assignments, and show that solution times for the implicit problem formulation are significantly lower than for the explicit formulation.

The remainder of the methods in the categorization are heuristic methods. In decomposition methods, the problem is divided into smaller subproblems that are easier to solve and the solution for the original problem is obtained by combining solutions from subproblems. Solving subproblems might require additional assumptions and heuristics. Decomposition approaches rely on the specific structures of the problem that can be utilized. Becker [49] uses a heuristic decomposition approach for rotational workforce scheduling, where first a fixed set of shift blocks covering staffing requirements is created in the master problem and then a feasible sequence of shift blocks is generated in the subproblem.

In goal programming, a set of targets is placed and the deviations from these predetermined target levels are minimized in the objective function. Todovic et al. [50] set targets for monthly working hours of police officers and the number of police officers in daily and nightly shifts in their application of goal programming to staff scheduling of a police station in Bosnia and Herzegovina. In the working set generation approach, only a subset of feasible solutions is inspected and used to return the final solution. Bechtold and Brusco [51] review papers where this approach is applied.

The construction and improvement category contains algorithms where new employees are iteratively added to the schedule until all constraints are met. Lü and Hao [35] propose a sophisticated adaptive neighborhood search algorithm for a highly constrained nurse scheduling instance. The next category is metaheuristics. Metaheuristics are higher-level procedures that require little assumptions on the specific problem instance and therefore are applied in a variety of domains. Specifically, genetic algorithms and tabu search have been widely applied in workforce scheduling, but also other methods such as iterative local search and particle swarm optimization have seen use in literature [1]. LP-based methods are discussed in a separate section and the other category contains all approaches that do not belong to any of the listed categories. In these cases, the selection of the solution method is usually related to a unique scheduling situation.

2.5.1 Relaxation based heuristics

A straightforward approach for solving integer programming problems is to ignore integer constraints and solve a relaxed version of the problem with an established linear or nonlinear optimization method. A heuristic is applied to the relaxed solution and variables with fractional parts are restored to integer values. A near-optimal solution is not guaranteed, and the feasibility of the solution depends on the specifics of the rounding heuristic.

In personnel scheduling, Dantzig [22] proposed rounding as a way to treat situations where solving the problem results in fractional values for the number of workers starting a shift. However, a systematic procedure was not provided. Later authors have suggested systematic procedures for the same problem formulation. Keith [52], in a study of automated scheduling of operators at Illinois Bell Telephone Company, starts by rounding relaxed decision variables with fractional value to the closest integer. After that, for each tour it is calculated how much adding a new employee would reduce understaffing. After that, the employee assignment which maximally reduces understaffing is added to the solution. If multiple assignments reduce understaffing by the same amount, then the increase of the sum of squared overstaffing is calculated for each candidate assignment, and the assignment causing the smaller increase of overstaffing is selected. The automatic schedule provided a better fit to the forecasted workload than previously used manual schedules.

Shoewalter and Mabert [19] study four rounding heuristics. Three of the heuristics are based on rounding fractional values down, and one is based on rounding up. After that, new employees are added to or removed from shifts while minimizing overstaffing and maximizing the reduction of understaffing using different heuristic-specific rules, until all the workload is covered. In general, heuristics based on rounding down and adding new employees produced better results than the heuristic based on rounding up.

Bartholdi [53] develops a rounding heuristic for cyclic tour scheduling problems. In cyclic problems, possible tours share the same structure: all schedules are constructed from one schedule by rotating assignments in the tour, i.e., moving them one period forward. Bartholdi proves that the proposed heuristic has a better upper bound than the naive rounding up of fractional values. Bartholdi also proposed a generalized version of the heuristic for noncyclic tour scheduling problems.

Li et al. [54] applies three rounding heuristics inspired by [19] and [53] for a staff scheduling of a lockbox system of a commercial bank. Special skill requirements for the work and extreme fluctuation of demand characterize the problem. The heuristics produced nearly optimal results and computational times on all instances were acceptable. Li [54] concludes that studying heuristic solution methods is well justified.

Cezik, Gunluk, and Luss [55] model weekly tour scheduling, inspired by call centers, as an integer programming problem. The relaxed problem instance without integer constraints is solved and variables with zero value are fixed. For variables with a large fractional part, a new constraint is introduced that forces the variable to a higher value than its current value, and the relaxed problem is solved again.

Steps are repeated while variables with large fractional parts are left. The remaining variables are optimized with a branch-and-bound procedure with a time limit. Half of the studied problem instances are solved optimally by the heuristic.

Fowler, Wirojanagud, and Gel [26] minimize hiring, training, salary, firing, and missed production costs. Two LP-based heuristics are proposed where decision variables related to hiring, training, and firing are rounded up or down based on which option worsens the objective value less. The performance of the heuristics is compared to a genetic algorithm, a decomposition algorithm, and naive rounding heuristics. Two LP-based algorithms are shown to produce near-optimal results in computation time that is just a fraction of the time of the best-performing genetic algorithm.

This review has shown that personnel scheduling provides a rich and widely studied problem field, where many objectives are pursued, a variety of solution methods are applied, and various constraints are present. The model discussed in subsequent chapters incorporates aspects from models briefly explored in this chapter – a multiobjective days-off scheduling model for heterogeneous employees, differing both on their skills and contracts, is formulated. The model is solved by constructing an integer solution with a heuristic from a real number solution for a relaxed problem instance without integer constraints.

3 The personnel scheduling model

This chapter introduces a model for a multiobjective days-off scheduling problem for a multi-skilled workforce with varying competencies in each skill and employee-specific contracts.

3.1 Problem setting

The goal of the problem is to assign rest and workdays for employees inside the planning period such that the workload forecast is matched with a minimal roster. The planning period consists of whole weeks and its length can be anything from 1 to 26 weeks. The workload is separated into task types in the model. Employee's competence for every task type is modelled with a reward parameter. An employee can perform work on a task type only if they have a nonzero reward. Workload forecast in the model is deterministic and given on a day-task type precision.

The model has four objectives. A match between employees' skills and an estimate on what tasks types employees spend their shifts is rewarded. Idle work, workdays on weekends, and deviations from the desired employee-specific ideal workweek length are penalized. In addition, objective functions are defined such that the amount of rewarded work and the amount of idle work are balanced between days. Similarly, the number of weekend workdays and deviations from the ideal workweek length is balanced between employees. The objective functions are discussed in detail in section 3.3.

Matching workload task types and employee skills requires estimating which task types an employee works on during a day, and how much time is spent on each assignment. Therefore, the problem is similar to a tour scheduling problem. However a proper shift schedule with breaks and start times for each task is not provided, and therefore, the problem is in fact a days-off scheduling problem.

An important concept in the model is idle work, which is work time that is not assigned to any task. In general, idle work is undesirable but it is needed to fulfill employee-level minimum work constraints if there is not enough workload.

Employees have four types of days: rest days, unallocated days, fixed days and locked days. On a rest day employee does not perform work but a rest day may still be treated as a workday in constraints. An unallocated day does not have information whether an employee is working or having a rest day. A fixed day is a workday, but it does not have information on what task types an employee works on during the day. A locked day is a workday that contains assignments to task types which can not be modified. The decision for every unallocated day, whether an employee is working or resting, is the output of the model.

Additionally, employees have employee-specific limits for minimum and maximum amounts of work. Those units that are used for limits, parameters and variables are weeks, days, quarters – which equals fifteen minutes and describes granularity of task assignments –, and minutes.

3.2 Sets, parameters, and decision variables

Table 1 contains sets of the model, Table 2 contains parameters of the model, and Table 3 contains decision variables of the model.

Table 1: Sets of the personnel scheduling model and their explanations.

Label	Explanation	Element
E	Employees.	e
D	Days in the planning period. $\|D\|$ is the length of the planning period	d
D_w	Weekend days in the planning period. Fridays, Saturdays, and Sundays are considered weekend days.	d
K	Weeks in the planning period.	k
T	Task types.	t
F	Fixed days for employees. A fixed day is a workday for which assignments to tasks types during the day are not specified.	(e, d)
L	Locked days for employees. A locked day is a workday for which assignments to tasks types during the day are known.	(e, d)
R	Rest days for employees. A rest day is a day that does not contain any idle work or work that is assigned to task types. Rest days may still affect constraints limiting the amount of work for an employee: for example, sick leaves do not contain active work but are still treated as workdays in constraints.	(e, d)
U	Unallocated days for employees. An unallocated day is a day that does not yet have a decision whether the employee is working or not.	(e, d)
U_e	Unallocated days for employee e. Defined as $U_e = \{d \mid (e, d) \in U\}$.	d

3.3 Objective functions

The objective function, which is formed from four components and soft constraints, is minimized to obtain a high quality schedule of rest days and workdays for employees. The first objective function, task score f_1, measures how well shift assignments are aligned with employees' skills. Task score is defined as

$$f_1 = \sum_{d \in D} \left(1 - \frac{\sum_{e \in E} \sum_{t \in T} (reward_{e,t} \cdot x_{e,d,t})}{max\_reward \cdot total\_workload_d} \right)^2 . \tag{1}$$

Table 2: Parameters of the personnel scheduling model and their explanations.

Label	Explanation	Unit of time
min_weekly_e	The minimum weekly balance for employee e.	Minutes
max_weekly_e	The maximum weekly balance for employee e.	Minutes
min_total_e	The minimum balance for employee e during the planning period.	Minutes
max_total_e	The maximum balance for employee e during the planning period.	Minutes
$max\_weekly\_days_e$	The maximum number of workdays during the week for employee e.	Days
$max\_total\_days_e$	The maximum number of workdays for employee e during the planning period.	Days
$max\_consecutive\_days_e$	The maximum number of consecutive workdays for employee e.	Days
$base\_duration_e$	The workday length for the employee e for unallocated days that are set to the workday.	Quarters
$reward_{e,t}$	A weight depicting how competent employee e is in task type t. The reward is in the range [0, 12].	-
max_reward	The highest possible reward, equals 12.	-
$balance_{e,d}$	The amount of time that should be considered for employee e on day d in constraints that are defined in minutes. Not necessarily the same as the work time of the day.	Minutes
$is\_workday_{e,d}$	A Boolean parameter defining whether a day is taken into account in constraints that are defined in days.	-
$worktime_{e,d}$	The length of the workday of a fixed or a locked day. Might differ from the $base\_duration_e$.	Quarters
$shift\_assignment_{e,d,t}$	The amount of time spent on task type t on a locked day (e, d).	Quarters
$workload_{d,t}$	The forecasted demand on day d for task t that is not covered by locked days and have to be covered by assigning work on fixed and unallocated days.	Quarters
$total\_workload_d$	The total estimated workload over all tasks during day d. Includes also the workload covered by shifts assignments of locked days.	Quarters

Table 3: Decision variables of the personnel scheduling model and their explanations.

Label	Explanation	Domain for values
$x_{e,d,t}$	Shift assignments, i.e., quarters worked by employee e on day d on task type t. In practice values for variables are integers but for the purposes of this days-off scheduling problem approximating shift assignments with real numbers is adequate.	$\mathbb{R}_{\geq 0}$
$y_{e,d}$	1 if employee e is working on day d and 0 if e is not working. Defined only for unallocated days.	$\{0,1\}$

The quadratic form of the objective function aims to balance the distribution of rewarded work between days. A situation where work is distributed evenly between days is preferred to a situation where some days are left with very little work in comparison to other days. The same idea of using quadratic terms to balance distribution is applied also in other objective functions.

The second objective function is idle penalty f_2, which penalizes idle work, i.e., worktime that is not assigned to any particular task type. Idle work does not have a corresponding workload and is not rewarded in task score. It can be written as

$$idle\_work_{e,d} = \begin{cases} y_{e,d} \cdot base\_duration_e - \sum_{t \in T} x_{e,d,t} & \forall\,(e,d) \in U \\ worktime_e - \sum_{t \in T} x_{e,d,t} & \forall\,(e,d) \in F \cup L \\ 0 & \forall\,(e,d) \in R \end{cases} \tag{2}$$

and the objective function is defined as

$$f_2 = \sum_{d \in D} \left(\frac{\sum_{e \in E} idle\_work_{e,d}}{total\_workload_d} \right)^2 . \tag{3}$$

In the model, each employee has a goal for the number of workdays during the week. The goal is derived from the minimum weekly balance reflecting the general goal of building a minimal roster. The target workweek length for employee e is

$$weekly\_target_e = \left\lceil \frac{min\_weekly_e}{base\_duration_e} \right\rceil . \tag{4}$$

The third objective function, workweek length penalty f_3, penalizes deviations from the ideal workweek length. Maximum deviation from the target is written as

$$max\_deviation_e = \max(weekly\_target_e, 7 - weekly\_target_e). \tag{5}$$

Let

$$workday_{e,d} = \begin{cases} \mathbf{1}_{is\_workday_{e,d}} & (e,d) \in F \cup L \cup R \\ y_{e,d} & (e,d) \in U \end{cases} \tag{6}$$

where $\mathbf{1}_X$ is an indicator function:

$$\mathbf{1}_X = \begin{cases} 1 & X \\ 0 & \text{not } X. \end{cases} \tag{7}$$

The workweek length penalty f_3 is defined as

$$f_3 = \sum_{e \in E} \left(\frac{1}{|K|} \sum_{k \in K} \left(\frac{\sum_{d \in k} workday_{e,d} - weekly\_target_e}{max\_deviation_e} \right)^2 \right). \tag{8}$$

The model gives special emphasis on workdays to weekends and Fridays. The fourth objective function, weekend work penalty f_4, penalizes weekend days workdays:

$$f_4 = \sum_{e \in E} \left(\frac{\sum_{d \in D_w} workday_{e,d}}{|D_w|} \right)^2. \tag{9}$$

Furthermore, soft constraints are included in the objective function. Let

$$g_1 = \sum max\_weekly\_slack_{e,k} \tag{10}$$
$$g_2 = \sum max\_total\_slack_e \tag{11}$$
$$g_3 = \sum min\_weekly\_slack_{e,k} \tag{12}$$
$$g_4 = \sum min\_total\_slack_e \tag{13}$$
$$g_5 = \sum consecutive\_days\_slack_{e,d}, \tag{14}$$

where $max\_weekly\_slack_{e,k}$, $max\_total\_slack_e$, $min\_weekly\_slack_{e,k}$ $min\_total\_slack_e$ and $consecutive\_days\_slack_{e,d}$ are slack variables that are discussed in section 3.4. Objective functions and soft constraints are combined to form the final objective function f, written as

$$f = \sum_{i=1}^{4} \alpha_i f_i + \sum_{j=1}^{5} \beta_j g_j, \tag{15}$$

where α_i and β_j are weights defining the relative importance of objectives and soft constraints.

3.4 Soft constraints

Employees have contractual constraints that define the minimum and the maximum amount of worktime inside a workweek or a planning period, and a parameter for the maximum amount of consecutive workdays. These parameters are treated as soft constraints in the model. Constraints from the input data are preprocessed to take into account existing manual allocations. The need for a preprocessing step has been noted in the literature by Mirrazavi [56].

3.4.1 Maximum weekly balance

The maximum amount of work for an employee during week is limited with two parameters in the model: max_weekly_e defines maximum work time in minutes for the employee and $max\_weekly\_days_e$ defines maximum number of workdays during week. The constraint is only created for those employee-week combinations for which the employee has unallocated days. New employee-week specific maximum limits are calculated by subtracting existing manually set balance consumption from max_weekly_e and rounding it down, and by subtracting existing manually set workdays from $max\_weekly\_days_e$, and finally setting the more strict of these max limits as the final mas limit. Formally:

$$
\begin{aligned}
max\_weekly_{e,k} &= \left\lfloor \frac{max\_weekly_e - \sum_{d \in k \setminus U_e} balance_{e,d}}{15 \cdot base\_duration_e} \right\rfloor \\
max\_weekly\_days_{e,k} &= max\_weekly\_days_e - \sum_{d \in k \setminus U_e} \mathbf{1}_{is\_workday_{e,d}} \\
max\_weekly_{e,k} &= \min(max\_weekly_{e,k}, max\_weekly\_days_{e,k}). \qquad (16)
\end{aligned}
$$

If $max\_weekly_{e,k} \leq 0$ then maximum limit was already reached by manual allocations and every unallocated day inside the week is allocated to rest day, and no constraint is created for the week. For remaining weeks, the constraint is written as

$$max\_weekly\_slack_{e,k} = max(0, \sum_{d \in k \cap U_e} y_{e,d} - max\_weekly_{e,k}). \qquad (17)$$

3.4.2 Maximum total balance

Handling the maximum limit for the total amount of work during the planning period is analogous to implementing a maximum weekly balance limit. The constraint is only created for those employees that have unallocated days in the planning period. Maximum limits are given both in minutes in max_total_e and in days in $max\_total\_days_e$. The more strict alternative is selected as the final limit after rounding max_total_e:

$$
\begin{aligned}
max\_total_e &= \left\lfloor \frac{max\_total_e - \sum_{d \in D \setminus U_e} balance_{e,d}}{15 \cdot base\_duration_e} \right\rfloor \\
max\_total\_days_e &= max\_total\_days_e - \sum_{d \in D \setminus U_e} \mathbf{1}_{is\_workday_{e,d}} \\
max\_total_e &= \min(max\_total_e, max\_total\_days_e) \qquad (18)
\end{aligned}
$$

If $max\_total_e \leq 0$ then maximum limit was already reached by manual allocations and every unallocated day in planning period is allocated to a rest day and no constraint is created. For remaining employees constraint is written as

$$max\_total\_slack_e = max(0, \sum_{d \in U_e} y_{e,d} - max\_total_e). \qquad (19)$$

3.4.3 Minimum weekly balance

The minimum amount of work minutes for employee e during any week is given by the min_weekly_e parameter. The constraint is only created for employee-week combinations for which the employee has unallocated days. For each week k in the planning period, a new employee-week specific minimum limit is calculated by subtracting existing manually set balance consumption. The limit is rounded up to ensure that the preprocessed limit is expressed in full days, and that the limit is not looser than the limits in the input data:

$$min\_weekly_{e,k} = \left\lceil \frac{min\_weekly_e - \sum_{d \in k \setminus U_e} balance_{e,d}}{15 \cdot base\_duration_e} \right\rceil . \tag{20}$$

If $min\_weekly_{e,k} \leq 0$ minimum balance limit is already reached and constraint is not needed in the model. For remaining active constraints is set

$$min\_weekly_{e,k} = \min(min\_weekly_{e,k}, max\_weekly_{e,k}) \tag{21}$$

to ensure that the minimum limit is always lower than the maximum limit. The constraint is written as

$$min\_weekly\_slack_{e,k} = max(0, min\_weekly_{e,k} - \sum_{d \in k \cap U_e} y_{e,d}). \tag{22}$$

3.4.4 Minimum total balance

Handling the minimum limit for the total amount of work during the planning period is analogous to implementing the minimum weekly balance limit. The constraint is only created for those employees that have unallocated days in the planning period. A new employee-specific minimum limit is calculated by subtracting the existing manually set balance consumption. The limit is rounded up to ensure that the preprocessed limit is expressed in full days, and that the limit is not looser than the limits in the input data:

$$min\_total_e = \left\lceil \frac{min\_total_e - \sum_{d \in D \setminus U_e} balance_{e,d}}{15 \cdot base\_duration_e} \right\rceil . \tag{23}$$

If $min\_total_e \leq 0$ then the minimum limit is already reached and the constraint is not needed for the employee. For remaining employees with an active total minimum limit is ensured that the minimum limit is always lower than the maximum limit:

$$min\_total_e = \min(min\_total_e, max\_total_e) \tag{24}$$

The constraint is written as

$$min\_total\_slack_e = max(0, min\_total_e - \sum_{d \in U_e} y_{e,d}). \tag{25}$$

3.4.5 Maximum consecutive days

The parameter $max\_consecutive\_days_e$ gives the maximum amount of consecutive workdays. Data outside the planning period is considered: if there exists a long streak of workdays right after or right before the current planning period, extending it to cover over $max\_consecutive\_days_e$ workdays is undesired. The constraint is needed only for those sequences of days that consist of workdays and unallocated days, and at least one of the days is an unallocated day. Let

$$period_{e,d} = \{d, d+1, ..., d + max\_consecutive\_days_e\} \tag{26}$$

and let Boolean variables

$$no\_rest\_days_{e,d} = \sum_{i \in period_{e,d}} \mathbf{1}_{(e,i)\notin U \text{ and } \neg is\_workday_{e,i}} = 0 \tag{27}$$

$$unallocated\_days_{e,d} = \sum_{i \in period_{e,d}} \mathbf{1}_{(e,i)\in U} > 0. \tag{28}$$

Let

$$D^* = \{1 - max\_consecutive\_days_e, \dots, |D|\}. \tag{29}$$

The set of employee-days that need a constraint can be written as

$$\{(e,d) \in E \times D^* \mid no\_rest\_days_{e,d} \text{ and } unallocated\_days_{e,d}\} \tag{30}$$

and for employee-days in this set the constraint is written as

$$consecutive\_days\_slack_{e,d} = \tag{31}$$

$$\max(0, \sum_{d \in period_{e,d} \cap U_e} y_{e,d} + \sum_{d \in period_{e,d} \setminus U_e} 1 - max\_consecutive\_days_e). \tag{32}$$

In this formulation, possible long workday streaks that break $max\_consecutive\_days_e$ with many days are recorded as many separate violations of one day instead of one violation of many days.

3.5 Hard constraints

In addition to worktime-related soft constraints, for which violations are minimized, the model has hard constraints that are always fulfilled.

3.5.1 Manual shift assignments

Shift assignments set by a planner are not modified:

$$x_{e,d,t} = shift\_assignment_{e,d,t} \qquad \forall\, (e,d,t) \in L \times T \tag{33}$$

$$x_{e,d,t} = 0 \qquad \forall\, (e,d,t) \in R \times T. \tag{34}$$

3.5.2 Skill requirement

Assigning tasks is possible only for employees with nonzero skill:

$$x_{e,d,t} = 0 \qquad \forall\, d \in D \text{ if } reward_{e,t} = 0. \tag{35}$$

3.5.3 Workday length constraint

The sum of work over all tasks for an employee on any given day has maximum limit. For unallocated days the limit is $base\_duration_e$. For fixed days the limit is $worktime_e$. The constraint is written as:

$$\sum_{t \in T} x_{e,d,t} \leq y_{e,d} \cdot base\_duration_e \qquad \forall\, (e,d) \in U \tag{36}$$

$$\sum_{t \in T} x_{e,d,t} \leq worktime_e \qquad \forall\, (e,d) \in F. \tag{37}$$

3.5.4 Coverage constraint

The amount of work allocated for a task on a given day can not exceed the amount of open workload for that task:

$$\sum_{e \in \{e | (e,d) \in F \cup U\}} x_{e,d,t} \leq workload_{d,t} \qquad \forall\, (d,t) \in D \times T. \tag{38}$$

This type of constraint, which allows only understaffing, is the rarest type of coverage constraint in the literary review by Van den Bergh et al. [1]. In this model, it is a design choice that supports the goal of always building a minimal roster.

4 Heuristic algorithms for the model

This chapter introduces two solution heuristics for the model introduced in chapter 3. Both heuristics construct an integer solution from the optimal solution of the relaxed problem, where the workday variables $y_{e,d}$ are real numbers in the range $[0, 1]$. The solution for the relaxed nonlinear optimization problem is obtained with Ipopt solver [57].

The heuristics always respect maximum limits but meeting minimum limits is not guaranteed. This is often the desired prioritization by planners. Additionally, heuristics are designed with a moderate running time in mind, as it is vital for their usability in practice.

4.1 Fix-and-Optimize heuristic

A straightforward general method for constructing an integer solution from a real number solution, is to fix part of the variables to a integer value by rounding them to the nearest integer, and reoptimize the remaining variables with fractional value. This is repeated until every variable is rounded to integer value. The first heuristic, Fix-and-Optimize, applies this idea to the problem discussed in this thesis.

In the Fix-and-Optimize heuristic those workday variables that are sufficiently close to zero are fixed to zero, variables sufficiently close to one are fixed to one, and the remaining variables along with all shift assignment variables are reoptimized with Ipopt. Thresholds for rounding to zero and one are updated, and the routine is repeated. Thresholds themselves are based on empirical testing. The iterations are repeated until the thresholds for fixing variables to zero and one match. For every candidate for fixing to one, maximum constraint violations are calculated. If fixing would breach any of the maximum limits, then the workday variable is fixed to zero instead. The pseudocode for the heuristic is shown in Algorithm 1.

Algorithm 1 Fix-and-Optimize heuristic

Let $LB = [0.1, 0.2, 0.4, 0.6]$
Let $UB = [0.9, 0.8, 0.7, 0.6]$
for $lb, ub \in zip(LB, UB)$ **do**
  Solve relaxed problem to update $y_{e,d}$ and $x_{e,d,t}$
  for $(e, d) \in U$ **do**
    if $y_{e,d} < lb$ **or** Fixing to one breaches max limit **then**
      Add constraint $y_{e,d} = 0$
      Remove (e, d) from U
    else if $y_{e,d} \geq ub$ **then**
      Add constraint $y_{e,d} = 1$
      Remove (e, d) from U
    end if
  end for
end for
Workday variables are integers, reoptimize $x_{e,d,t}$

4.2 Consolidate-Idle heuristic

Inside a workweek, idle work can be reallocated freely without affecting weekly or total balance constraints. This observation forms the basis for the second heuristic called Consolidate-Idle.

As the first step, the heuristic sets all workday variables that are sufficiently close to 0 or 1. 0.01 is used as a threshold for proximity. Every time a workday is fixed, it is ensured that no maximum limits are breached. If fixing it to one breaches a maximum limit, the workday variable is fixed to zero instead. As a second step, for every week that has more than one unallocated employee-day, idle work is moved from a day with the smallest workday variable value to a day with the largest value. If the receiving day cannot take more idle work due to the workday length constraint, the remaining idle work is allocated to other workdays in the week, in descending order with respect to the workday variable value. These moves are applied until no reallocations are available. Next, workday variables close to one or zero are fixed similarly to the first step. After reallocating idle work, for every week that has not yet reached the minimum weekly limit, unallocated days are fixed to one starting from the day with the highest workday variable value, until the minimum limit is met or no more unallocated days are left for the week. A similar fixing is performed for every employee until all total minimum limits are reached or the employee does not have any more unallocated days in the planning period. The remaining unallocated employee-days are fixed greedily to one or zero depending on which is more favorable for the objective function. The pseudocode for the heuristic is shown in Algorithm 2.

Algorithm 2 Consolidate-Idle heuristic

Solve relaxed problem
Fix every $y_{e,d}$ that is close to one or zero and remove (e, d) from U
while Reallocations available **do**
    Move idle work inside week from the day with the lowest value for $y_{e,d}$ to the day with the highest value for $y_{e,d}$ that is not one
end while
Fix every $y_{e,d}$ that is close to one or zero and remove (e, d) from U
while Weeks with unsatisfied minimum limits **do**
    Fix day with the highest $y_{e,d}$ inside week to one
    Remove (e, d) from U
end while
while Employees with unsatisfied minimum total limits **do**
    Fix day with highest $y_{e,d}$ to one inside the planning period for employee
    Remove (e, d) from U
end while
for $(e, d) \in U$ **do**
    Round $y_{e,d}$ to one or zero depending which is better for the objective
    Remove (e, d) from U
end for
Workday variables are integers, reoptimize $x_{e,d,t}$

5 Computational results

This chapter introduces the data set which is used to evaluate heuristics. Additionally, comparisons on solution times, objective function values, and the feasibility of solutions obtained with heuristics are presented. Computations were carried out on a Dell Latitude 7490 laptop running on an Intel Core i5-8250U processor, with Windows 10 Pro as an operating system. The algorithms were implemented with Python v. 3.9.9 using Pyomo framework v. 6.0.1, and the version of Ipopt was 3.12.13. The algorithms were not limited by memory.

5.1 Data set

The heuristics were tested using a comprehensive data set of 60 instances. The test instance with label test_49 in Table A1 had no unallocated days or shift assignment variables after constraint preprocessing and therefore it was excluded from the set. All figures and graphs are based on the 59 remaining test instances. Figure 1 shows the distribution of instances against metrics describing the size of the instance.

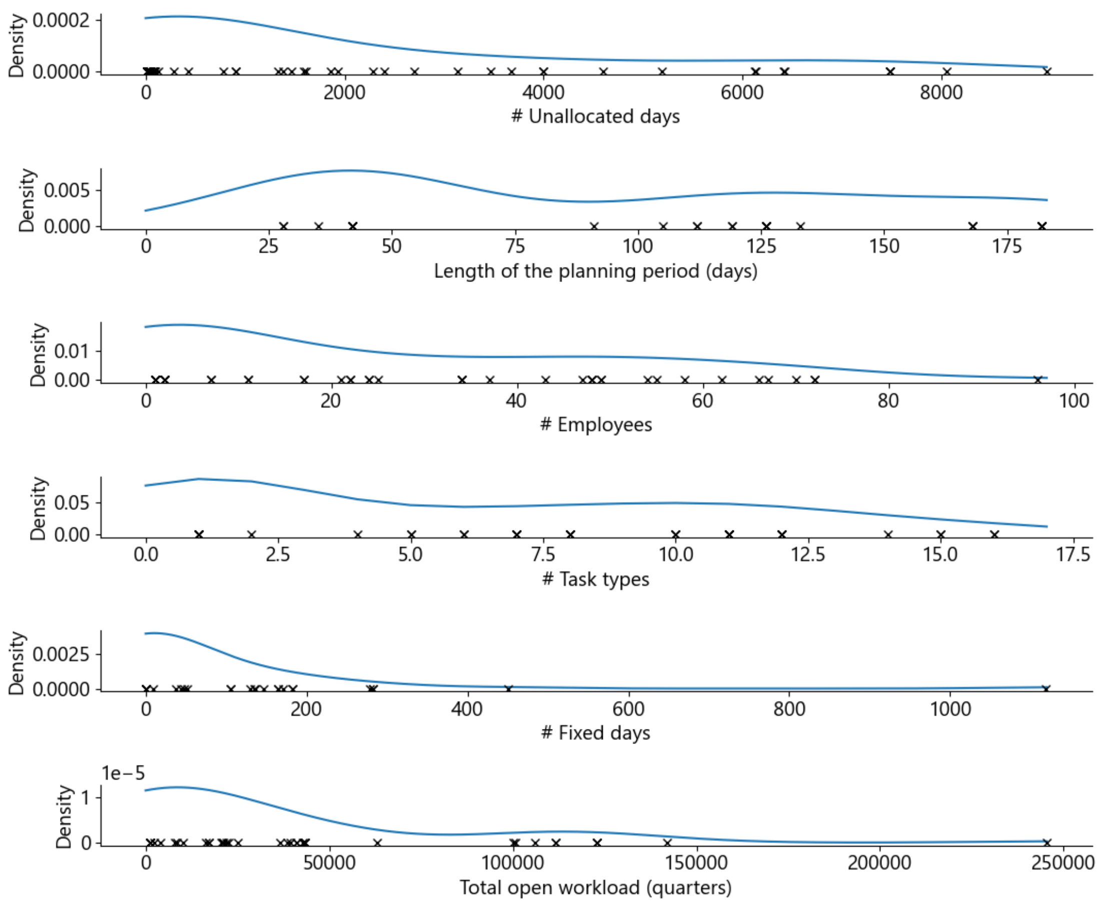

Figure 1: A Gaussian kernel density estimate and a rug plot of instances with respect to the selected instance attributes.

5.2 Solution times

Figure 2 shows the number of individual solved instances in a given time. Values are presented on both a linear and a logarithmic time scale for heuristics and the time for solving the original instance without integer constraints. The graphs show that obtaining a real number solution for the first time is the most important component affecting the performance of the heuristics and it also sets a minimum limit for their solution times.

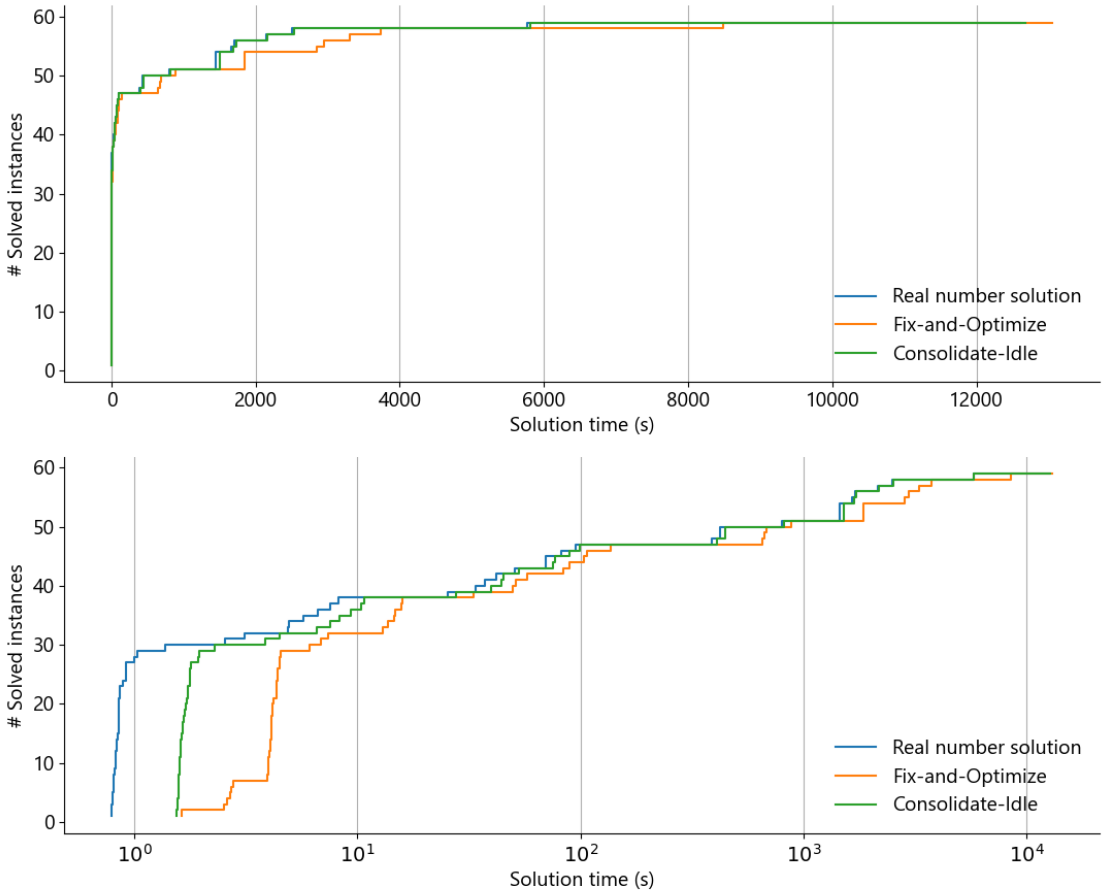

Figure 2: Number of instances solved as a function of the solution time.

The sum of solution times over the whole data set for Consolidate-Idle is 1.6% higher and for Fix-and-Optimize 31.2% higher than the sum of real number solution times. This comparison is heavily influenced by instances with long solution times. Dividing solution time of every instance with the real number solution time shows that the average solution time for the Fix-and-Optimize is 3.1 times longer than for the real number solution, the median being 2.7 times longer. For Consolidate-Idle, on average, the solution time is 1.5 times longer than for the real number solution, the median being 1.5 times longer. The differences in performance arise because Fix-and-Optimize calls a solver many times while Consolidate-Idle calls it only twice: once to get the real number solution and in last step to reoptimize shift assignment variables. Out of 59 instances, Consolidate-Idle was the fastest heuristic on 58 instances and Fix-and-Optimize had the best solution time once.

Solution times varied considerably. While most instances were solved under 100 seconds, the longest solution time was over 10000 seconds. Figure 3 shows how the solution times of the relaxed problem scale when they are compared against the number of unallocated employee-days, the number of constraints, the number of variables, and the sum of open workload. The solution times increase almost exponentially with the growing size of the problem.

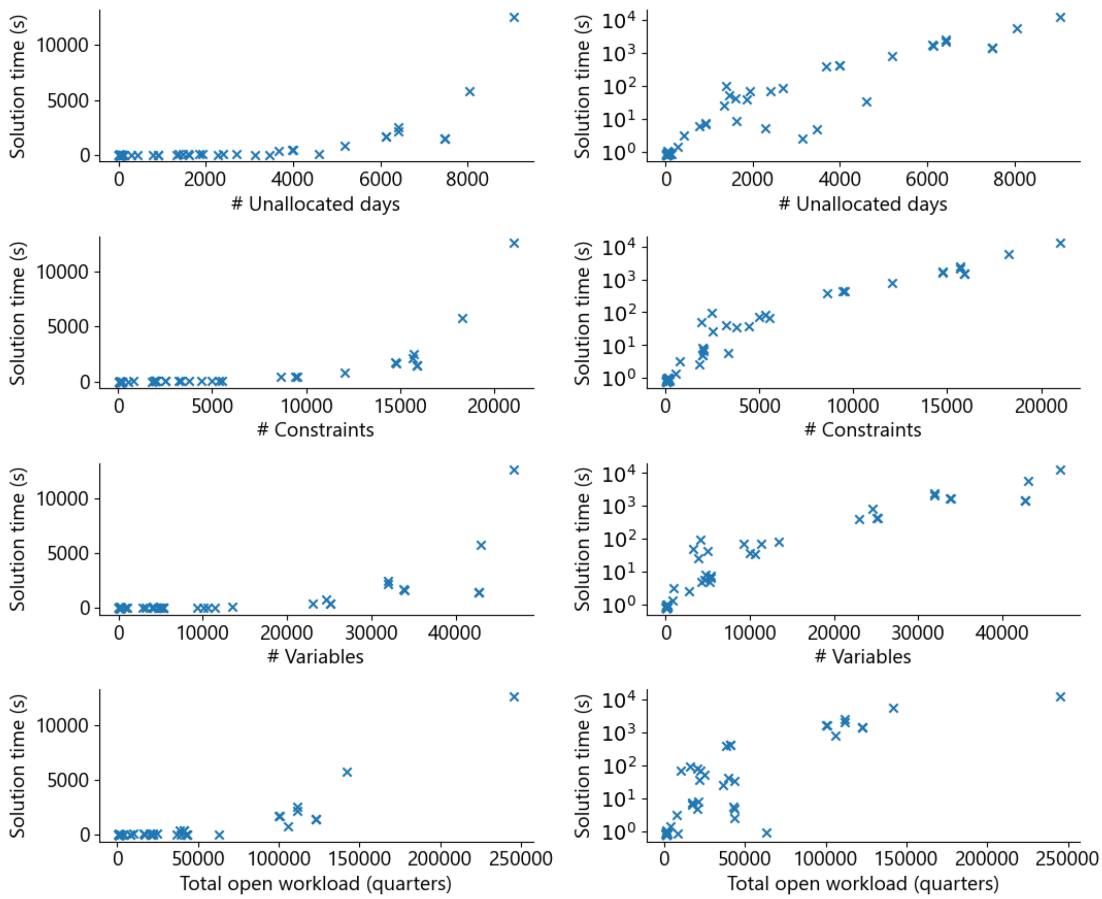

Figure 3: Solution time of the relaxed problem compared against metrics that describe the size of the problem. On the left the time scale is linear and on the right logarithmic.

The correlation to the solution time is clearest for the number of constraints and number of variables. A small number of unallocated employee-days does not always predict solution time well, because the distribution of manually allocated rest days, locked days, and fixed days significantly affects the resulting number of constraints and variables in the model. Similarly, a small open workload does not always predict the solution time well.

5.3 Quality of results

The data set contains instances of various sizes. Therefore, normalized objective functions with objective values in the range $[0,1]$ are used for analyzing results to make comparisons easier. Table 4 shows expressions for the normalized objective functions and unfulfilled workload. The unfulfilled workload is not part of the objective function but it is an important metric for assessing the quality of the solution.

Table 4: Normalized objective functions and unfulfilled workload.

Metric	Symbol	Normalized metric
Task score	f_1	$f_1/\|D\|$
Idle penalty	f_2	$f_2/\|D\|$
Week length penalty	f_3	$f_3/\|E\|$
Weekend work penalty	f_4	$f_4/\|E\|$
Total objective	$\sum_{i=1}^{4} \alpha_i f_i$	$\frac{\sum_{i=1}^{4} \alpha_i f_i}{\alpha_1 \|D\| + \alpha_2 \|D\| + \alpha_3 \|E\| + \alpha_4 \|E\|}$
Unfulfilled workload	-	$1 - \frac{\sum_{e \in E} \sum_{d \in D} \sum_{t \in T} x_{e,d,t}}{\sum_{d \in D} \sum_{t \in T} workload_{d,t}}$

Figure 4 shows the distribution of normalized objective function values and unfulfilled workload for all instances. The boxes in the graph extend from the lower quartile to the upper quartile of the data, with a line at the median. The whiskers extend a distance of 1.5 times the height of the box to both directions and points outside this range are shown separately. The closer the value is to zero the better it is. The zero objective function value in most cases is not obtainable due to missing skills, too few available workers, contract limitations, or other reasons.

The idle penalty is close to zero for most real number solutions. Fix-and-Optimize is able to keep solutions close to zero while Consolidate-Idle deviates more from zero. For multiple instances a large amount of idle work can not be avoided and they are shown as outliers in the box plot.

A visual inspection of total objective and task score shows rather similar distributions, which is explained by task scores having the highest weight in total objective. For real number solutions, the median of unfulfilled workload on many instances is close to the optimal, which corresponds to a situation where all forecasted workload is met with shift assignments. However, the whiskers cover the whole range of possible values, showing the existence of instances where no forecasted workload is met due to the structure of manual allocation or missing skills of employees. For Fix-and-Optimize and Consolidate-Idle, the median of unfulfilled workload is significantly higher because many days with a small amount of assigned work end up being rounded to rest days.

In Table 5, integer solutions from heuristics are compared to real number solutions. The mean, median, and standard deviation of the difference between the heuristics

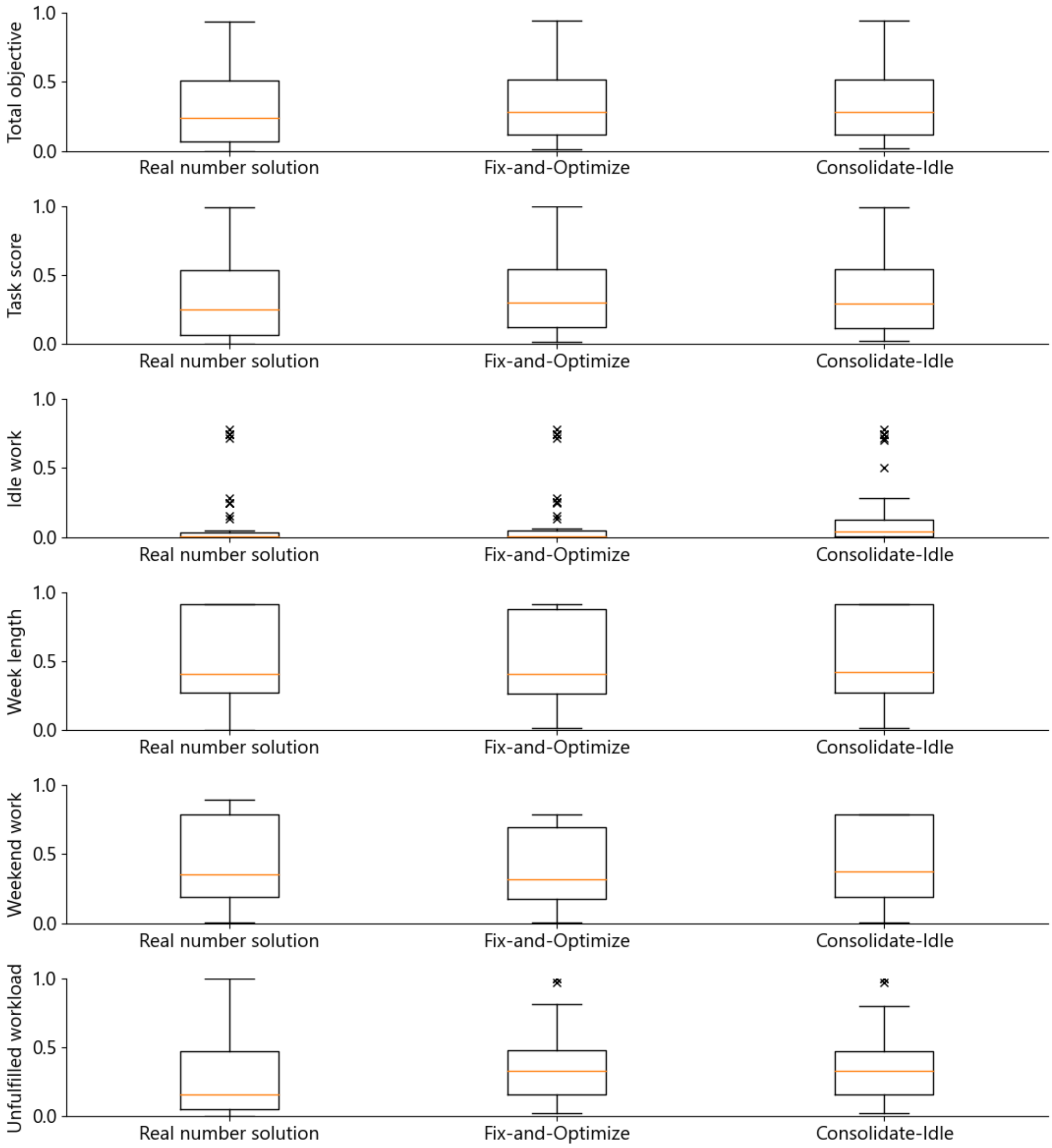

Figure 4: The total objective, objective functions and the amount of unfulfilled workload for real number solutions and integer solutions obtained with each heuristic.

and the real number solution are reported for each metric.

For Consolidate-Idle, the median of differences of total objective, task score, and unfulfilled workload is smaller than for Fix-and-Optimize. However, for Fix-and-Optimize the medians for the idle penalty, the week length penalty and the weekend work penalty are better than for Consolidate-Idle. The week length penalty and the weekend work penalty are lower for Consolidate-Idle than for real number solutions. Fix-and-Optimize performs well on the these metrics because it assigns rest days more aggressively than Consolidate-Idle, which is rewarded by these metrics.

Table 5: The mean, median and standard deviations of differences between integer solutions and the real number solution as percentages. The difference of metric's value between the integer solution and the real number solution is calculated for every instance and statistical figures are reported based on all instances. Negative value for mean or median indicates that metric's value is better for integer solutions and positive value indicates that metric's value is better for real number solutions. Best, i.e. lowest, means and medians are highlighted.

	Fix-and-Optimize			Consolidate-Idle		
	Mean	Median	Std.	Mean	Median	Std.
Total objective (%)	**3.8**	1.3	4.8	4.2	**0.7**	5.6
Task score (%)	**4.0**	1.4	5.0	4.2	**0.8**	5.7
Idle work (%)	**0.4**	**0.0**	0.8	3.5	0.5	9.4
Week length (%)	**-3.6**	**-0.5**	7.4	-1.6	0.3	7.8
Weekend work (%)	**-6.7**	**-3.3**	8.0	-4.6	-0.1	11.0
Unfulfilled workload (%)	5.8	2.2	7.4	**5.2**	**1.4**	7.4

On the other hand, means for the task score and total objective for Fix-and-Optimize are lower than for Consolidate-Idle. Figure 5 shows that in instances where total objectives of integer solutions are significantly larger than the total objective of the real number solution, Consolidate-Idle produces solutions that are farther from the real number solution. In the Figure, these instances are visible as points far from the origin and below the $y = x$ line. These points skew the mean such that the mean for Fix-and-Optimize becomes lower than for Consolidate-Idle, while the opposite is true for medians. Standard deviations for total objective for Consolidate-Idle is higher than for Fix-and-Optimize, which is caused by the same subgroup of instances.

Table 6 shows how many times Consolidate-Idle or Fix-and-Optimize produced the best integer result. Row totals may add up to more than 59 because both heuristics can end up with the same objective function value. The results are in line with Table 5. Consolidate-Idle produces more best results for the total objective and task score while Fix-and-Optimize produces more best results for other objectives.

Table 6: Number of best scores with respect for each objective for each heuristic.

	Fix-and-Optimize	Consolidate-Idle
# best total objectives	28	**31**
# best task scores	26	**34**
# best idle penalties	**58**	13
# best work week length penalties	**53**	23
# best weekend work penalties	**44**	29

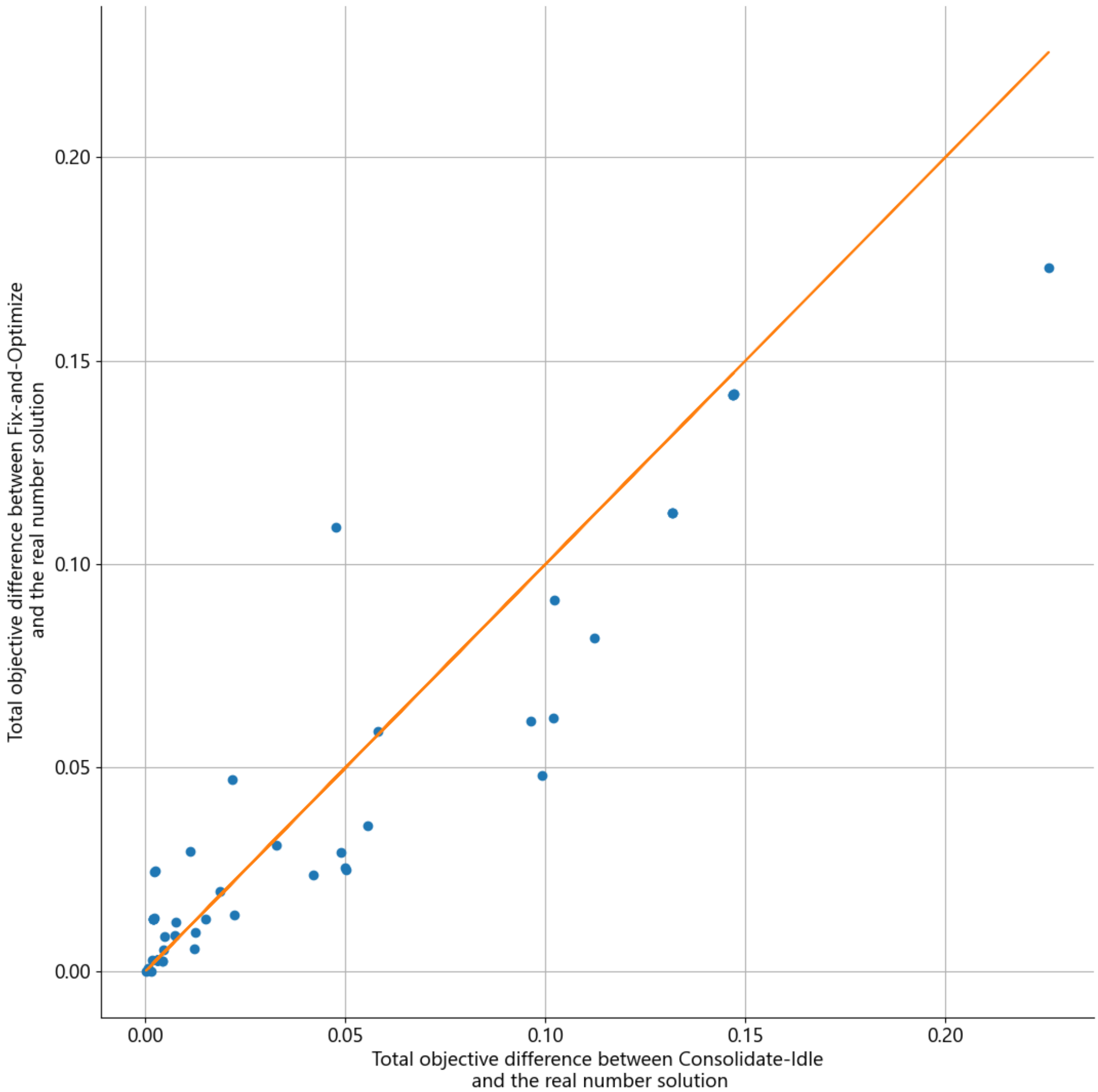

Figure 5: The total objective difference to the real number solution for Fix-and-Optimize and Consolidate-Idle with respect one another.

5.4 Feasibility of results

Table 7 shows the average percentage of violated soft constraints for an instance for each type of soft constraint. Violations were measured as the number of nonzero slack variables in the final solution.

All soft constraints related to the maximum number of workdays were respected by design. For minimum weekly constraints Fix-and-Optimize was six times more likely to violate the minimum weekly constraint than Consolidate-Idle. Violating the minimum total constraint was more common than violating weekly minimum. Fix-and-Optimize violated the minimum total constraint five times more often than Consolidate-Idle. The difference between the heuristics is expected, as Fix-and-Optimize does not consider minimum limits at all, while Consolidate-Idle has a step

Table 7: Average percentage of violated soft constraints for an instance. The smallest share of minimum constraints violated is highlighted.

	Fix-and-Optimize	Consolidate-Idle
Min weekly work	1.8	**0.3**
Min total work	30.8	**6.5**

in which as many minimum limits are fulfilled as possible.

Figure 6 shows the difference of the normalized total objective between Fix-and-Optimize and Consolidate-Idle with respect to the difference of the share of violated total minimum and weekly minimum constrains for each instance. Consolidate-Idle did not produce a solution that breaches more minimum limits than Fix-and-Optimize in any of the instances. In addition, when Fix-and-Optimize produced a solution with a lower total objective than Consolidate-Idle, it at the same time violated more minimum constraints.

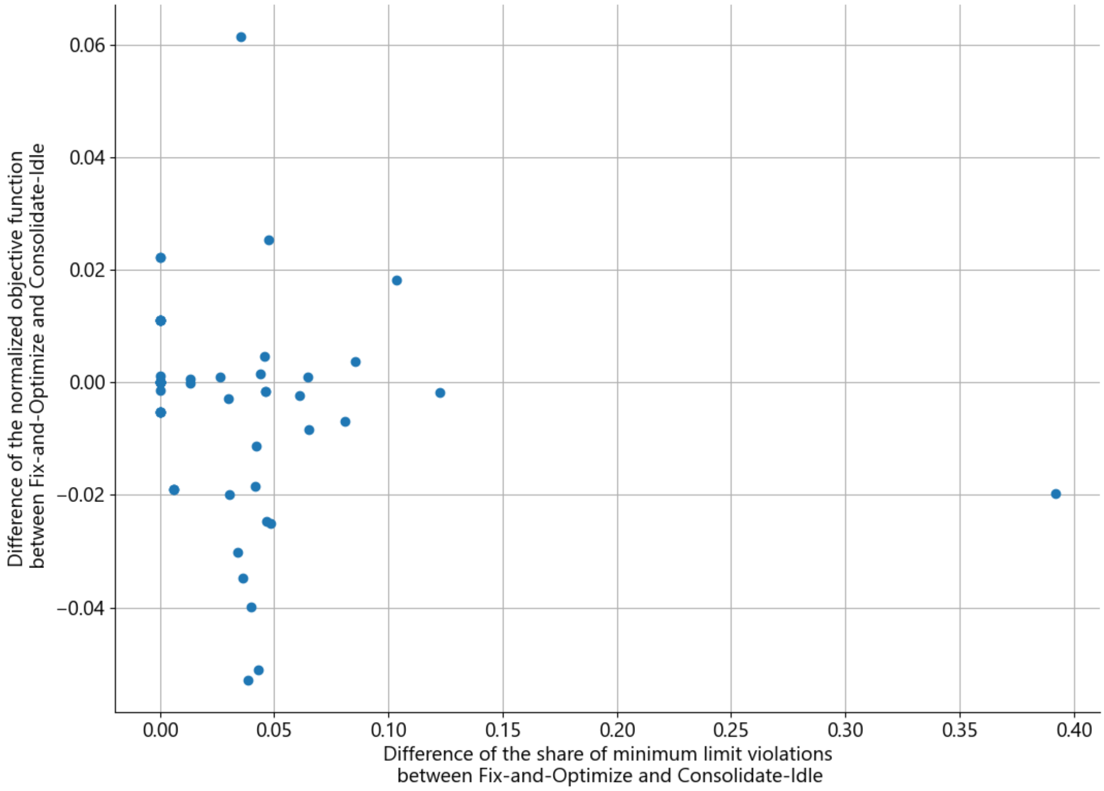

Figure 6: The difference of normalized total objective between Fix-and-Optimize and Consolidate-Idle solutions with respect to the difference of the share of violated minimum limits.

6 Conclusions

In this study, performant methods for obtaining a good-quality solution for a particular days-off scheduling model were investigated. It was shown that constructing an integer solution from the solution of a relaxed problem with a heuristic is an efficient method to create high-quality solutions for the studied model. Out of the two studied heuristics, Consolidate-Idle required less time for solving instances and violated less soft constraints. Regarding objective function values, results were mixed. Fix-and-Optimize provided more often the best objective function values on three objective functions of four, while Consolidate-Idle provided best results on one objective function and total objective. However, it was shown that when Fix-and-Optimize produced a solution with better total objective, it also at the same time violated more soft minimum constraints.

The main bottleneck for performance is the solution time of the real number solution. In future research, decomposition approaches, experimenting with different solvers, and using a feasible, but not the optimal, real number solution obtained with lighter computational requirements as a starting point for heuristics, are possible method-focused ways for improving solution times. Regarding modelling, finding alternative ways to formulate constraints, or reducing the number of constraints with preprocessing or heuristic steps before solving the model, are also topics of interest as with test instances a large number of constraints lead to a long solution time. In addition, the implementations of heuristics were not optimized in this study, leaving room for performance gains especially with instances with short solution times.

The heuristics presented in this study were conceptually simple. Investigating whether more sophisticated rounding rules improve objective function values of the final solution remains an interesting open question. Also, the selection of weights for objective functions and soft constraints, a topic ignored in this thesis, is important for the quality of the solution obtained.

It must be noted, that from the point of view of a worker, many aspects affect the goodness of the schedule: forecast quality, unexpected events such as sick leaves of other workers, the quality of shift assignments during the day, and the timing and the amount of rest during the day. Therefore, the experience of the worker, and the total cost, and other benefits of the automated schedule can be estimated only to a limited extent from a mere days-off schedule. A more unified view is needed together with testing in real-life working environments to fully evaluate the quality of automatic scheduling.

References

[1] Van den Bergh, Jorne and Beliën, Jeroen and De Bruecker, Philippe and Demeulemeester, Erik and De Boeck, Liesje. *Personnel scheduling: A literature review.* European journal of operational research, vol. 226, no. 3, pp. 367–385, 2013.

[2] Burke, Edmund K and De Causmaecker, Patrick and Berghe, Greet Vanden and Van Landeghem, Hendrik. *The state of the art of nurse rostering.* Journal of scheduling, vol. 7, no. 6, pp. 441–499, 2004.

[3] Defraeye, Mieke and Van Nieuwenhuyse, Inneke. *Staffing and scheduling under nonstationary demand for service: A literature review.* Omega, vol. 58, pp. 4–25, 2016.

[4] Montreuil, Benoit and Poulin, Marc. *Demand and supply network design scope for personalized manufacturing.* Production Planning & Control, vol. 16, no. 5, pp. 454–469, 2005.

[5] Gopalakrishnan, Balaji and Johnson, Ellis L. *Airline crew scheduling: State-of-the-art.* Annals of Operations Research, vol. 140, no. 1, pp. 305–337, 2005.

[6] Heil, Julia and Hoffmann, Kirsten and Buscher, Udo. *Railway crew scheduling: Models, methods and applications.* European Journal of Operational Research, vol. 283, no. 2, pp. 405–425, 2020.

[7] Nissen, Volker and Günther, Maik. *Automatic generation of optimised working time models in personnel planning.* International Conference on Swarm Intelligence, pp. 384–391, 2010.

[8] Baker, Kenneth R. *Workforce allocation in cyclical scheduling problems: A survey.* Journal of the Operational Research Society, vol. 27, no. 1, pp. 155–167, 1976.

[9] Ernst, Andreas T and Jiang, Houyuan and Krishnamoorthy, Mohan and Sier, David. *Staff scheduling and rostering: A review of applications, methods and models.* European journal of operational research, vol. 153, no. 1, pp. 3–27, 2004.

[10] Brusco, Michael J and Johns, T Reid. *The effect of demand characteristics on labour scheduling methods.* International Journal of Operations & Production Management, 1995.

[11] Franz, Lori S and Miller, Janis L. *Scheduling medical residents to rotations: solving the large-scale multiperiod staff assignment problem.* Operations Research, vol. 41, no. 2, pp. 269–279, 1993.

[12] Bailey, James and Field, John. *Personnel scheduling with flexshift models.* Journal of operations management, vol. 5, no. 3, pp. 327–338, 1985.

[13] Rong, Aiying. *Monthly tour scheduling models with mixed skills considering weekend off requirements.* Computers & Industrial Engineering, vol. 59, no. 2, pp. 334–343, 2010.

[14] Henao, César Augusto and Muñoz, Juan Carlos and Ferrer, Juan Carlos. *The impact of multi-skilling on personnel scheduling in the service sector: a retail industry case.* Journal of the Operational Research Society, vol. 66, no. 12, pp. 1949–1959, 2015.

[15] Fırat, M and Briskorn, Dirk and Laugier, Alexandre. *A branch-and-price algorithm for stable workforce assignments with hierarchical skills.* European Journal of Operational Research, vol. 251, no. 2, pp. 676–685, 2016.

[16] Walter, Matthias and Zimmermann, Jürgen. *On a multi-project staffing problem with heterogeneously skilled workers.* Operations Research Proceedings 2011, pp. 489–494, 2012.

[17] Wang, Zhaojing and Hu, Hao and Gong, Jie. *Competence Based Worker Assignment and Impacts on Production Scheduling in Precast Construction.* 2018 IEEE Technology and Engineering Management Conference (TEMSCON), pp. 1–6, 2018.

[18] Topaloglu, Seyda. *A shift scheduling model for employees with different seniority levels and an application in healthcare.* European Journal of Operational Research, vol 198, no. 3, pp. 943–957, 2009.

[19] Showalter, Michael J. and Mabert, Vincent A. *An evaluation of a full-/part-time tour scheduling methodology.* International Journal of Operations & Production Management, 1988.

[20] Mabert, Vincent A. and Showalter, Michael J. *Measuring the impact of part-time workers in service organizations.* Journal of Operations Management, vol. 9, no. 2, pp. 209–229, 1990.

[21] Valls, Vicente and Pérez, Ángeles and Quintanilla, Sacramento. *Skilled workforce scheduling in service centres.* European Journal of Operational Research, vol. 193, no. 3, pp. 791–804, 2009.

[22] Dantzig, George B. *Letter to the editor—A comment on Edie's "Traffic delays at toll booths".* Journal of the Operations Research Society of America, vol. 2, no. 3, pp. 339–341, 1954.

[23] Di Gaspero, Luca and Gärtner, Johannes and Kortsarz, Guy and Musliu, Nysret and Schaerf, Andrea and Slany, Wolfgang. *The minimum shift design problem.* Annals of operations research, vol. 155, no. 1, pp. 79–105, 2007.

[24] Pastor, Rafael and Olivella, Jordi. *Selecting and adapting weekly work schedules with working time accounts: A case of a retail clothing chain.* European Journal of Operational Research, vol. 184, no. 1, pp. 1–12, 2008.

[25] Ingels, Jonas and Maenhout, Broos. *The impact of overtime as a time-based proactive scheduling and reactive allocation strategy on the robustness of a personnel shift roster.* Journal of Scheduling, vol. 21, no. 2, pp. 143–165, 2018.

[26] Fowler, John W and Wirojanagud, Pornsarun and Gel, Esma S. *Heuristics for workforce planning with worker differences.* European Journal of Operational Research, vol. 190, no. 3, pp. 724–740, 2008.

[27] Joubert, JW and Conradie, DG. *A fixed recourse integer programming approach towards a scheduling problem with random data: A case study.* ORiON, vol. 21, no. 1, pp. 1–11, 2005.

[28] Nasir, Jamal Abdul and Kuo, Yong-Hong. *A decision support framework for home health care transportation with simultaneous multi-vehicle routing and staff scheduling synchronization.* Decision Support Systems, vol. 138, pp. 113361, 2020.

[29] Al-Yakoob, Salem M and Sherali, Hanif D. *A column generation approach for an employee scheduling problem with multiple shifts and work locations.* Journal of the Operational Research Society, vol. 59, no. 1, pp. 34–43, 2008.

[30] Alsheddy, Abdullah and Tsang, Edward PK. *Empowerment scheduling for a field workforce.* Journal of Scheduling, vol. 14, no. 6, pp. 639–654, 2011.

[31] Shahnazari-Shahrezaei, Parisa and Tavakkoli-Moghaddam, Reza and Kazemipoor, Hamed. *Solving a new fuzzy multi-objective model for a multi-skilled manpower scheduling problem by particle swarm optimization and elite tabu search.* The International Journal of Advanced Manufacturing Technology, vol. 64, no. 9, pp. 1517–1540, 2013.

[32] Finco, Serena and Battini, Daria and Delorme, Xavier and Persona, Alessandro and Sgarbossa, Fabio. *Workers' rest allowance and smoothing of the workload in assembly lines.* International Journal of Production Research, vol. 58, no. 4, pp. 1255–1270, 2020.

[33] Ayough, Ashkan and Zandieh, M and Farsijani, H. *GA and ICA approaches to job rotation scheduling problem: considering employee's boredom.* The International Journal of Advanced Manufacturing Technology, vol. 60, no. 5, pp. 651–666, 2012.

[34] Xu, Shuling and Hall, Nicholas G. *Fatigue, Personnel Scheduling and Operations: Review and Research Opportunities.* European Journal of Operational Research, 2021.

[35] Lü, Zhipeng and Hao, Jin-Kao. *Adaptive neighborhood search for nurse rostering.* European Journal of Operational Research, vol. 218, no. 3, pp. 865–876, 2012.

[36] Bard, Jonathan F. *Selecting the appropriate input data set when configuring a permanent workforce.* Computers & Industrial Engineering, vol. 47, no. 4, pp. 371–389, 2004.

[37] Elshafei, Moustafa and Alfares, Hesham K. *A dynamic programming algorithm for days-off scheduling with sequence dependent labor costs.* Journal of Scheduling, vol. 11, no. 2, pp. 85–93, 2008.

[38] Hochbaum, Dorit S and Levin, Asaf. *Cyclical scheduling and multi-shift scheduling: Complexity and approximation algorithms.* Discrete Optimization, vol. 3, no. 4, pp. 327–340, 2006.

[39] Gärtner, Johannes and Musliu, Nysret and Slany, Wolfgang. *A heuristic based system for generation of shifts with breaks.* International Conference on Innovative Techniques and Applications of Artificial Intelligence, pp. 95–106, 2004.

[40] White, Christine A and Nano, Emilina and Nguyen-Ngoc, Diem-Hang and White, George M. *An evaluation of certain heuristic optimization algorithms in scheduling medical doctors and medical students.* International Conference on the Practice and Theory of Automated Timetabling, pp. 105–115, 2006.

[41] Maghzi, Parisa and Roohnavazfar, Mina and Mohammadi, Mohammad and Naderi, Bahman. *A Mathematical Model for Operating Room Scheduling Considering Limitations on Human Resources Access and Patient Prioritization.* Journal of Quality Engineering and Production Optimization, vol. 4, no. 2, pp. 67–82, 2019.

[42] Guerriero, Francesca and Guido, Rosita. *Modeling a flexible staff scheduling problem in the Era of Covid-19.* Optimization Letters, pp. 1–21, 2021.

[43] Zucchi, Giorgio and Iori, Manuel and Subramanian, Anand. *Personnel scheduling during Covid-19 pandemic.* Optimization Letters, vol. 15, no. 4, pp. 1385–1396, 2021.

[44] Alfares, Hesham K. *Survey, categorization, and comparison of recent tour scheduling literature.* Annals of Operations Research, vol. 127, no. 1, pp. 145–175, 2004.

[45] Alfares, Hesham K. *Staffing and workforce scheduling for a security gate.* International Journal of Operations and Quantitative Management, vol. 7, no. 4, pp. 281–294, 2001.

[46] Yaoyuenyong, Kriangchai and Nanthavanij, Suebsak. *Energy-based workforce scheduling problem: mathematical model and solution algorithms.* ScienceAsia, vol. 31, pp. 383-393, 2005.

[47] Zhu, Xiaomei and Sherali, Hanif D. *Two-stage workforce planning under demand fluctuations and uncertainty.* Journal of the Operational Research Society, vol. 60, no. 1, pp. 94–103, 2009.

[48] Bechtold, Stephen E. and Jacobs, Larry W. *Implicit modeling of flexible break assignments in optimal shift scheduling.* Management Science, vol. 36, no. 11, pp. 1339-1351, 1990.

[49] Becker, Tristan. *A decomposition heuristic for rotational workforce scheduling.* Journal of Scheduling, vol. 23, no. 5, pp. 539–554, 2020.

[50] Todovic, Dragana and Makajic-Nikolic, Dragana and Kostic-Stankovic, Milica and Martic, Milan. *Police officer scheduling using goal programming.* Policing: An International Journal of Police Strategies & Management, 2015.

[51] Bechtold, Stephen E. and Brusco, Michael J. *Working set generation methods for labor tour scheduling.* European Journal of Operational Research, vol. 74, no. 3, pp. 540–551, 1994.

[52] Keith, Elbridge Gerry. *Operator scheduling.* AIIE Transactions, vol. 11, no. 1, pp. 37–41, 1979.

[53] Bartholdi III, John J. *A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering.* Operations Research, vol. 29, no. 3, pp. 501–510, 1981.

[54] Li, Cheng and Robinson Jr, E. Powell and Mabert, Vincent A. *An evaluation of tour scheduling heuristics with differences in employee productivity and cost.* Decision Sciences, vol. 22, no. 4, pp. 700–718, 1991.

[55] Cezik, Tolga and Günlük, Oktay and Luss, Hanan. *An integer programming model for the weekly tour scheduling problem.* Naval Research Logistics (NRL), vol. 48, no. 7, pp. 607–624, 2001.

[56] Mirrazavi, S Keyvan and Beringer, Henri. *A web-based workforce management system for Sainsburys Supermarkets Ltd.* Annals of Operations Research, vol. 155, no. 1, pp. 437–457, 2007.

[57] Wächter, Andreas and Biegler, Lorenz T. *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming.* Mathematical programming, vol. 106, no. 1, pp. 25–57, 2006.

A Characteristics of the test data set

Table A1 shows the number of employees, number of days (i.e. length of the planning period), number of tasks, number of unallocated employee-day pairs, and number of fixed employee days for every instance in the data set.

Table A1: Characteristic numbers of the test data set.

Label	$\lvert E\rvert$	$\lvert D\rvert$	$\lvert T\rvert$	$\lvert U\rvert$	$\lvert F\rvert$
data_0	11	126	6	905	283
data_1	1	91	15	10	0
data_2	96	133	8	92	0
data_3	1	42	1	25	0
data_4	67	182	4	5190	0
data_5	17	119	7	1615	136
data_6	62	28	15	1380	44
data_7	55	182	11	9053	130
data_8	48	182	8	6121	182
data_9	11	126	6	905	283
data_10	1	42	1	41	0
data_11	1	42	1	18	0
data_12	72	182	10	7478	0
data_13	47	182	12	8047	0
data_14	2	42	1	76	0
data_15	70	182	10	7478	0
data_16	2	42	1	84	0
data_17	72	182	10	7478	0
data_18	48	182	8	6131	182
data_19	2	42	1	84	0
data_20	49	182	12	6422	105
data_21	2	42	1	76	0
data_22	1	42	1	42	0
data_23	1	42	1	34	0
data_24	34	168	12	3674	147
data_25	17	126	7	1931	0
data_26	1	42	1	42	0
data_27	34	168	11	3991	164
data_28	21	126	5	2394	0

data_29	22	126	7	1859	0
data_30	7	105	2	126	0
data_31	34	168	11	3993	164
data_32	1	42	1	42	0
data_33	1	42	1	18	0
data_34	48	126	11	3132	135
data_35	1	42	1	42	0
data_36	24	126	7	2288	450
data_37	1	42	1	17	0
data_38	66	126	11	4599	279
data_39	49	182	12	6415	105
data_40	1	42	1	42	0
data_41	22	126	5	2700	10
data_42	54	126	11	3465	171
data_43	1	42	1	41	0
data_44	2	42	1	76	0
data_45	1	42	1	33	0
data_46	37	112	16	1335	48
data_47	24	28	8	430	38
data_48	1	42	1	41	0
data_49	43	112	16	1594	52
data_50	58	35	1	1465	0
data_51	7	42	1	282	0
data_52	25	119	14	779	1119
data_53	1	42	1	41	0
data_54	21	28	1	0	0
data_55	1	42	1	18	0
data_56	1	42	1	42	0
data_57	1	42	1	18	0
data_58	1	42	1	42	0
data_59	1	42	1	19	0