UNIVERSITY OF TURKU

AN ALGORITHMIC APPROACH TO SHIFT STRUCTURE OPTIMIZATION

Nico Kyngäs

TURUN YLIOPISTON JULKAISUJA – ANNALES UNIVERSITATIS TURKUENSIS
SARJA – SER. F OSA – TOM. 21 — TECHNICA – INFORMATICA — TURKU 2023

University of Turku

Faculty of Technology
Department of Computing
Computer Science
Doctoral Programme in Technology

Supervised by

Adj. Prof. Kimmo Nurmi
Faculty of Technology, University of Turku, Finland

Prof. Em. Olli Nevalainen
Faculty of Technology, University of Turku, Finland

Prof. Jukka Heikkonen
Faculty of Technology, University of Turku, Finland

Reviewed by

Prof. Jeroen Belien
Faculty of Economics and Business, KU Leuven, Belgium

Prof. Martti Juhola
Faculty of Information Technology and Communication Sciences, Tampere University, Finland

Opponent

Prof. Jan Westerholm
Faculty of Science and Engineering, Åbo Akademi, Finland

The originality of this publication has been checked in accordance with the University of Turku quality assurance system using the Turnitin OriginalityCheck service.

ISBN 978-951-29-9168-6 (PDF)
ISSN 2736-9390 (PRINT)
ISSN 2736-9684 (ONLINE)
Turku, Finland, 2023

UNIVERSITY OF TURKU
Faculty of Technology
Department of Computing
Computer Science
KYNGÄS, NICO: An Algorithmic Approach to Shift Structure Optimization
Doctoral dissertation, 69 pp.
Doctoral Programme in Technology
February 2023

ABSTRACT

Workforce scheduling in organizations often consists of three major phases: workload prediction, shift generation, and staff rostering. Workload prediction involves using historical behaviour of e.g. customers to predict future demand for work. Shift generation is the process of transforming the determined workload into shifts as accurately as possible. In staff rostering, the generated shifts are assigned to employees. In general the problem and even its subproblems are NP-hard, which makes them highly challenging for organizations to solve. Heuristic optimization methods can be used to solve practical instances within reasonable running times, which in turn can result in e.g. improved revenue, improved service, or more satisfied employees for the organizations. This thesis presents some specific subproblems along with practical solution methods.
KEYWORDS: heuristic optimization, personnel task scheduling, shift design, shift generation, shift minimization, shift scheduling, shift structure optimization, staff rostering, workforce scheduling

TURUN YLIOPISTO
Teknillinen tiedekunta
Tietotekniikan laitos
Tietojenkäsittelytiede
KYNGÄS, NICO: An Algorithmic Approach to Shift Structure Optimization
Väitöskirja, 69 s.
Teknologian tohtoriohjelma
Helmikuu 2023

TIIVISTELMÄ

Työvoiman aikataulutusprosessi koostuu kolmesta päävaiheesta: työtarpeen ennustaminen, työvuorojen muodostus ja työvuorojen miehitys. Tulevaa työtarvetta ennustetaan pääasiassa menneisyyden asiakaskäytöksen perusteella käyttäen esimerkiksi tilastollisia malleja tai koneoppimiseen perustuvia menetelmiä. Työvuorojen muodostuksessa tehdään työvuororakenne, joka noudattaa ennustettua ja ennalta tiedettyä työtarvetta mahdollisimman tarkasti. Työvuorojen miehityksessä määritetään työvuoroille tekijät. Jokainen vaihe itsessään on haasteellinen ratkaistava. Erityisesti työvuorojen miehitys on yleensä NP-kova ongelma. On kuitenkin mahdollista tuottaa käytännöllisiä ratkaisuja järkevässä ajassa käyttäen heuristisia optimointimenetelmiä. Näin on saavutettavissa mitattavia hyötyjä mm. tuottoon, asiakkaiden palvelutasoon sekä työntekijöiden työtyytyväisyyteen. Tässä väitöskirjassa esitellään eräitä työvoiman aikataulutuksen aliongelmia sekä niihin sopivia ratkaisumenetelmiä.
ASIASANAT: heuristinen optimointi, työvuorojen koostaminen, työvuorojen muodostus, työvuorojen optimointi, työvuorolistan optimointi, työvuororakenteen optimointi

Acknowledgements

First and foremost I would like to thank my supervisor, Adj. Prof. Kimmo Nurmi, for his tireless support and general prodding. This thesis would not exist in any form without him.

I would like to thank my supervisor, Prof. Em. Olli Nevalainen, and my research director, Prof. Jukka Heikkonen, for their invaluable input. I would like to acknowledge my co-authors, Jari Kyngäs and Dries Goossens, for their breadth of experience and insight.

Lastly, I would like to thank my family for supporting me, my friends for listening to my venting, and my favorite musicians for keeping me sane.

15/02/2023
Nico Kyngäs

Table of Contents

Abbreviations

2RH	A ruin and recreate heuristic developed for the GTSGP
ESMPTSP	Extended Shift Minimization Personnel Task Scheduling Problem
FL	A set of SMPTSP benchmark instances by Fages & Lapègue [1]
GTSGP	General Task-based Shift Generation Problem
KEB	A set of SMPTSP benchmark instances by Krishnamoorthy et al. [2]
KN	A set of SMPTSP benchmark instances by Kyngäs & Nurmi [VI]
PEAST	Population, Ejection, Annealing, Shuffling, Tabu
PEASTP	Population, Ejection, Annealing, Shuffling, Tabu, Parallel
PTSP	Personnel Task Scheduling Problem
SMPTSP	Shift Minimization Personnel Task Scheduling Problem
SWMB	A set of SMPTSP benchmark instances by Smet et. al [3]

List of Original Publications

This dissertation is based on the following original publications, which are referred to in the text by their Roman numerals:

I Kyngäs, N., Goossens, D., Nurmi, K., & Kyngäs, J. (2012, April). Optimizing the unlimited shift generation problem. In European Conference on the Applications of Evolutionary Computation (pp. 508-518). Springer, Berlin, Heidelberg.

II ©2013 IEEE. Reprinted, with permission, from Kyngäs, N., Nurmi, K., & Kyngäs, J. (2013, April). Solving the person-based multitask shift generation problem with breaks. In 2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO) (pp. 1-8). IEEE.

III Nurmi, K., Kyngäs, N., & Kyngäs, J. (2019). Workforce Optimization: the General Task-based Shift Generation Problem. IAENG International Journal of Applied Mathematics, 49(4), pp. 393-400.

IV Kyngäs, N., Nurmi, K., & Goossens, D. (2019). The General Task-based Shift Generation Problem: Formulation and Benchmarks. In 9th Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2019) (pp. 301-319).

V Nurmi, K., & Kyngäs, N. (2021). A Successful Three-Phase Metaheuristic for the Shift Minimization Personnel Task Scheduling Problem. Advances in Operations Research, 2021.

VI Kyngäs, N., & Nurmi, K. (2021). The Extended Shift Minimization Personnel Task Scheduling Problem. Position and Communication Papers of the 16th Conference on Computer Science and Intelligence Systems, Annals of Computer Science and Information Systems, Vol. 26, 2021.

The list of original publications have been reproduced with the permission of the copyright holders.

1 Introduction

Workforce scheduling, also known as *staff scheduling*, *personnel scheduling*, and *labor scheduling*, is an important practical problem whose solution quality significantly affects an organization's finances, morale and quality of service [4]. It is a difficult and time-consuming problem faced by every organization that has employees working on shifts or on irregular working days.

Staff rostering, or assigning shifts and rest days to employees, has been a main focus of workforce scheduling studies. Less emphasis has been given to deciding the *shift structure* and the *tasks* and duties involved in particular shifts, also known as *shift generation*. Typical application domains of shift generation and staff rostering include hospitals, call centers, retail stores, home care, cleaning, manufacturing, guarding, and delivery of goods. In this thesis, application domains include call centers and hospitals.

The generated shifts are subsequently assigned to employees in staff rostering. Often days-off and sometimes vacations are also scheduled in staff rostering. Hard limits on working and resting times tend to be the most important constraints, since they are set in legally binding collective labor agreements and government regulations. From an operative and wellbeing viewpoint, employees' competences and preferences are also of high priority.

Fig. 1 shows the demand-oriented workforce scheduling process, as presented by Nurmi et al. [III]. The intuitive assumption to make is that shift generation has a higher impact on the final schedule than staff rostering. This is because shift generation is strictly an earlier part of the scheduling process, and as such its impact is carried throughout the subsequent process, including the staff rostering phase. This relative impact is examined in Section 2.3. This is the first time such analysis is conducted.

It follows that good algorithms for various shift generation problems are needed for efficient real-world solutions. A number of relevant shift generation problems and practical solution methods to those problems are presented in this thesis. Best known results on the well-known benchmark instances are produced.

Two new shift generation problems are introduced. Mathematical models and benchmark instances to these problems are presented. The models function as accurate problem descriptions, enabling future work on the problems. The models and benchmark instances together enable comparison with not only out-of-the-box com-

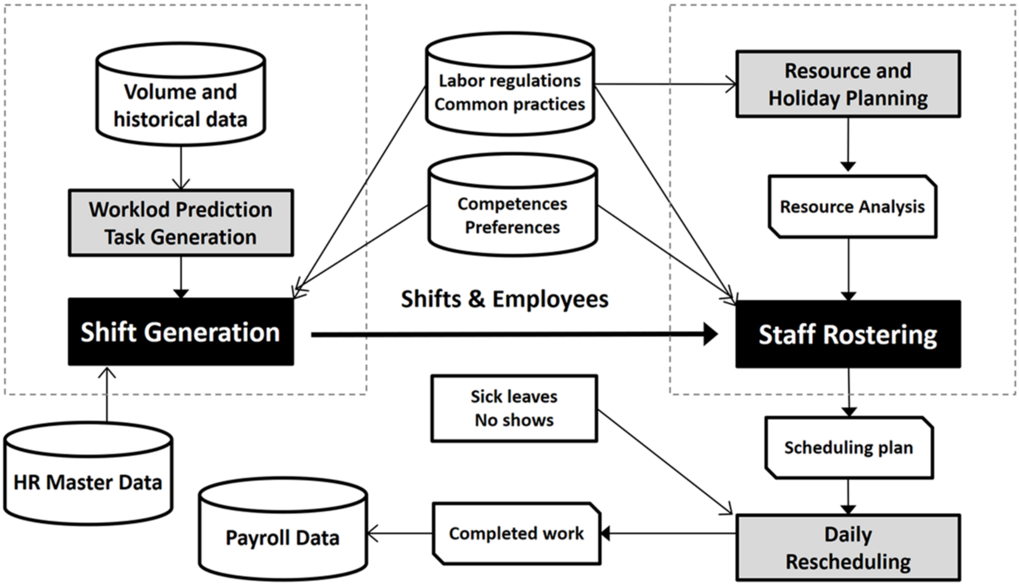

Figure 1. The demand-oriented workforce scheduling process, as presented by Nurmi et al. [III].

mercial solvers, but also any future approaches to the same problems.

The research questions addressed in this dissertation are as follows:

RQ1 Can employee-based demand be efficiently met with algorithmically generated shifts whose structure is not rigorously constrained?

RQ2 Can task-based demand be efficiently met with algorithmically generated shifts?

RQ3 Does shift structure optimization have significant effect on the achievable staff rosters?

The discipline of this thesis is primarily computer science, secondarily mathematics, and thirdly human science.

1.1 Publications

This thesis consists of six publications, the subjects and main contributions of which are as follows.

I: Optimizing the unlimited shift generation problem

In this work, a novel employee-based method for a single-day single-task shift generation problem is presented. An unlimited structure for individual shifts is allowed, while the objective guides both the global solution and the individual shifts towards promising regions of the search space. This is contrary to most of the solution methods to employee-based shift generation problems, in which the structure of an individual shift is usually highly constrained. The PEAST algorithm [5] is used to solve four real-world based benchmark instances. The presented solution approach successfully solves the problem.

II: Solving the person-based multitask shift generation problem with breaks

In this work, an employee-based method for a single-day multitask shift generation problem with breaks is presented. An almost completely free structure for individual shifts is allowed, while the objective guides both the global solution and the individual shifts towards promising regions of the search space. Multiple types of tasks and up to three breaks per shift are considered. This is contrary to most of the solution methods to employee-based shift generation problems, in which the structure of an individual shift is usually highly constrained. The PEAST algorithm is used to solve five real-world based benchmark instances. The presented solution approach successfully solves the problem.

III: Workforce Optimization: the General Task-based Shift Generation Problem

In this work, a rich task-based single-day shift generation problem, GTSGP, is introduced for the first time. The PEASTP algorithm is used to solve some instances of SMPTSP, which can be represented as a general task-based shift generation problem (GTSGP), and an artificial benchmark instance. The presented solution approach successfully solves the problem.

IV: The General Task-based Shift Generation Problem: Formulation and Benchmarks

In this work, a mathematical model for a task-based single-day shift generation problem, GTSGP, is introduced. A new benchmark generator is used to create 15 benchmark instances, and those are solved using the PEASTP algorithm and the general-purpose commercial solvers, Gurobi and CPLEX. The presented solution method is found to be superior in these tests.

V: A Successful Three-Phase Metaheuristic for the Shift Minimization Personnel Task Scheduling Problem

In this work, a new heuristic is developed for the shift minimization personnel task scheduling problem (SMPTSP). A constructive heuristic is used for generating quick initial solutions, which are then improved by a new ruin and recreate heuristic. Its result is fed to PEASTP to produce best known results on the well known SMPTSP benchmark instances from literature.

VI: The Extended Shift Minimization Personnel Task Scheduling Problem

In this work, the extended shift minimization personnel task scheduling problem (ESMPTSP), an extension to the SMPTSP, is introduced for the first time. The extension aims at maximizing the interchangeability of individual shifts between employees, yielding more flexible and robust shift structures whose resistance to e.g. sick leaves is improved. A ruin and recreate heuristic aimed at the more general problem of GTSGP is introduced and used to solve much simpler SMPTSP instances. First ESMPTSP benchmarks instances are also introduced and solved.

1.2 Contributions

Generic contributions of the author of this thesis are listed using CRediT [6], a system for classifying individual author contributions.

I: Optimizing the unlimited shift generation problem

Conceptualization: supporting; Methodology: equal; Software: lead; Validation: lead; Formal analysis: lead; Investigation: lead; Data curation: lead; Writing - original draft: equal; Writing - review & editing: equal; Visualization: lead.

The author implemented and adapted the PEAST algorithm to the unlimited shift generation problem.

II: Solving the person-based multitask shift generation problem with breaks

Conceptualization: equal; Methodology: equal; Software: lead; Validation: lead; Formal analysis: lead; Investigation: lead; Data curation: lead; Writing - original draft: equal; Writing - review & editing: equal; Visualization: lead.

The author explored different ways to handle breaks in unlimited shift generation problems and implemented one of them in the PEAST algorithm.

III: Workforce Optimization: the General Task-based Shift Generation Problem

Conceptualization: lead; Methodology: supporting; Validation: lead; Formal analysis: lead; Investigation: supporting; Data curation: supporting; Writing - original draft: supporting; Writing - review & editing: supporting; Visualization: supporting.

The author sought to introduce a task-based shift generation problem applicable to many different cases.

IV: The General Task-based Shift Generation Problem: Formulation and Benchmarks

Conceptualization: supporting; Methodology: lead; Software: lead; Validation: lead; Formal analysis: lead; Investigation: lead; Data curation: lead; Writing - original draft: equal; Writing - review & editing: equal; Visualization: lead.

The author formalized the GTSGP into a rigorous mathematical model and conducted computational experiments with Gurobi.

V: A Successful Three-Phase Metaheuristic for the Shift Minimization Personnel Task Scheduling Problem

Conceptualization: supporting; Methodology: equal; Software: equal; Validation: supporting; Formal analysis: lead; Investigation: supporting; Data curation:

supporting; Writing - original draft: equal; Writing - review & editing: equal; Visualization: lead.

The author implemented 2RH for the GTSGP and used it to conduct computational experiments.

VI: The Extended Shift Minimization Personnel Task Scheduling Problem

Conceptualization: supporting; Methodology: lead; Software: lead; Validation: equal; Formal analysis: equal; Investigation: equal; Data curation: supporting; Writing - original draft: equal; Writing - review & editing: equal; Visualization: lead.

The author formalized the ESMPTSP into a rigorous mathematical model and conducted computational experiments with 2RH and Gurobi.

2 Workforce Scheduling

The *demand-oriented workforce scheduling process*, as seen in Fig. 1 (Nurmi et al. [III]), starts from *workload prediction*, which is done based on both known and predicted events. For example, the arrival of customers can be predicted using collected data on patient flow in a hospital's intensive care unit, cash receipts in a supermarket, and received calls in a contact center. Known events may also be gathered from current sales contracts.

The workforce scheduling process continues with shift generation, which is essential to cost efficiency and service level in cases where the workload is not static. The most important goal in optimization is to cover the demand with as little shortage and surplus as possible. The generation of shifts is based on either a fluctuating demand on the number and type of employees, or the specific tasks that the shifts need to cover. In this work both cases are examined separately.

Future staffing requirements must be carefully considered in the *resource and holiday planning phase*. Holidays, training sessions, and other short-term absences impact staff rosters heavily, while forthcoming retirements and long-term sick leaves should also be considered. Resource availability and demand should be carefully considered and appropriate action taken to retain a chance of succeeding at matching the workforce with the shifts while adhering to the given constraints.

Even the best optimized rosters need to be changed. Daily *rescheduling* is necessary due to e.g. sick leaves and other no-shows. It is also possible for the demand to change after initial planning. Suitable substitute employees should be recommended considering the legal limitations, qualifications, employment contract, and salaries. The goal is usually to find the most economical legal candidates. Finally, the completed working times will be booked and made available for the payroll accounting system.

Workforce scheduling requires both optimization resources and human resources. The ultimate usefulness and utilization of the optimized shifts and rosters depend equally on the high quality produced in the preceding phases and the actual optimization result. When no shortcuts are taken in any of the phases, significant benefits in financial efficiency and employee satisfaction can be achieved. Solid overviews of workforce scheduling include those published by Musliu et al. [7], Ernst et al. [4], Di Gaspero et al. [8] and Van den Bergh et al. [9]. Kletzander and Musliu [10] gave a framework for the general employee scheduling problem.

The best results can theoretically be achieved by solving shift generation and staff rostering simultaneously. In practice, shift generation and staff rostering are often solved separately. Most variations of both shift generation and staff rostering are known to be NP-hard, see e.g. [11; 12; 13; 14; 15]. Nonetheless, some interesting implementations exist. Jackson et al. [16] presented a very simple randomized greedy algorithm that is light on hardware. Lapègue et al. [17] introduced the Shift Design and Personnel Task Scheduling Problem with Equity objective (SDPTSP-E). They constructed employee timetables by fixing days-off, designing shifts, and assigning fixed tasks within these shifts. Their objective was to minimize the number of unassigned tasks.

Dowling et al. [18] first created a master roster, a collection of working shifts and off shifts, and then allocated the tasks in the shifts. Çezik et al. [19] created daily shifts and cyclic weekly schedules out of them using a network flow based model. Prot et al. [20] first generate a set of interesting shifts. Then each shift is used to build a schedule by assigning tasks to employees. These two phases are iterated to improve solutions. They treated the constraint that each task must be assigned as a soft target instead. Smet et al. [21] presented the Integrated Task Scheduling and Personnel Rostering Problem, in which a set of fixed tasks is assigned to a set of employees. Each employee is assigned to a shift covering all their tasks that is selected from a pregenerated set of possible shifts.

2.1 Shift Generation

Shift generation transforms demand into shifts. The output includes the shift structure and potentially for each shift the break times, the tasks to be carried out, and the competences required. The generation of shifts is based on either a fluctuating demand on the number and type of employees, or the specific tasks that the shifts need to cover. In this thesis, these problems are separated into *employee-based* and *task-based* shift generation problems.

Many models and algorithms for employee-based shift generation problems have been developed. The first major contribution was published by Musliu et al. [7]. They proposed an employee-based problem, where the demand for a time period was given. The start times and the lengths of shifts was constrained, and an upper limit was given for the weekly average number of duties per employee. They generated solutions that minimized the number of different shifts, overstaffing, and understaffing, and balanced the average number of duties per week.

Bard & Wan [22] presented a cyclic multi-day multi-activity shift generation with breaks and transition costs between activities. A number of metaheuristic methods were developed to solve the problem. The methods were applied to real-world problem instances, and were found practically superior to direct CPLEX application to the problem model. Di Gaspero et al. [8] introduced an employee-based problem in which minimizing the number of different shifts was essential along with matching demand. Their schedules were cyclic, i.e. the last planning day of the planning horizon (e.g. one week) is immediately followed by the first planning day of the next cycle, and the requirements are repeated in each cycle. Acceptable shift types are characterized by earliest and latest start times, and minimum and maximum shift lengths.

Bhulai et al. [23] gave a generalized model for multi-skill shift generation in call centers. Their solution method generated a rough match between the predicted workload and available workforce, accounting for the stochasticity inherent in the call arrival process. Rekik et al. [24] considered a cyclic shift generation problem with fractionable breaks for a continuous 24-hour work day. A break can be broken into subbreaks as long as the length of each subbreak is a multiple of the timeslot length and adheres to min/max parameter values, the resulting work stretches adhere to min/max parameter values, and the total length of subbreaks is equal to the corresponding desired break length. Computational experiments show that the added flexibility of breaks leads to savings in required workforce.

Di Gaspero et al. [25] considered a cyclic one-week shift generation problem with breaks and the additional objective of minimizing the number of different shifts. A solution method employing local search to select shifts from the potential set of shifts, combined with a constraint programming model to determine breaks, was developed. Lequy et al. [26] introduced the multi-activity and task assignment problem (MATAP). The problem is basically shift generation with both employee-based demand and tasks combined. Due to the difficulty of the problem, a two-stage heuristic was proposed. In the first stage, a simplified problem was considered and only tasks are assigned to shifts. In the second stage, a variety of methods was used to assign demand-based activities to shifts.

Côté et al. [27] modeled a multi-activity shift generation and assignment problem. A binary relationship describes whether an employee can carry out a shift, enabling e.g. categorical skills and personal shift length limits to the model. A column generation approach based on context-free grammar was presented to solve the model. Brunner & Bard [28] presented a non-cyclic multi-day shift generation and assignment problem with breaks and both part-time and full-time employees. An employee's class affects the potential starting times and durations of the shifts assignable to the employee. A minimum ratio between full-timers and part-timers was used as a parameter to prevent the more flexible part-timer shifts from dominating the solution. A branch-and-price approach was developed to tackle the problem.

Boyer et al. [29] modeled a multi-activity shift generation and assignment problem with multiple skills and tasks. The tasks are fitted to timeslots like activities, and can have e.g. precedence constraints. The shifts are modeled using context-free grammar. The problems were solved using a branch-and-price scheme. Parisio & Neil Jones [30] solved a shift generation and assignment problem with uncertainty in staff demand. The constraints of the problem were split into staffing level (second stage) constraints, which ensure that a suitable number of skilled employees is used at all times, and other (first stage) constraints. The staffing constraints and costs were considered over a variety of forecast demand scenarios that were generated using a hidden Markov model and backward reduction algorithm. The objective was to minimize the cost of expected changes.

Dahmen & Rekik [31] presented a multi-day shift generation and assignment model with categorical skills. The model includes an upper bound on the total workforce budget, while the cost function consists of only overstaffing and understaffing. A tabu-inspired solution method with a variety of search memories was given to solve the problem. Lusby et al. [32] presented a one-week cardinality constrained shift design problem (CCSDP). The problem was to generate shifts while adhering to a maximum number of different shifts. No shifts spanning more than one day (i.e. shifts that start strictly before midnight and end strictly after midnight) were allowed, which allowed decomposition into daily problems. A matheuristic based on Benders decomposition was used to solve the problem.

Restrepo et al. [33] presented a multi-day multi-activity shift generation and assignment problem with multiple skills. The problem was solved in a branch-and-price framework, where a grammar-based approach was used as the pricing subproblem to constrain the search space of shifts. The master problem was solved with CPLEX. van Hulst et al. [34] presented a shift generation problem for multiple days that was solved using integer programming. The demand was considered an uncertain variable. The uncertainty was modelled as a part of the problem, resulting in a model the solutions of which were more robust with respect to changes in the demand.

Bonutti et al. [35] considered a real-world based cyclic multi-day multi-activity shift generation problem with at most a single break per shift. Additional objectives include minimization of the total number of different shifts and average shift length. A simulated annealing based metaheuristic method was developed to solve the problem. The method was tested on public real-world based benchmark instances introduced in the paper. Restrepo et al. [36] considered a discontinuous stochastic multi-day multi-activity shift generation and assignment problem with breaks. Context-free grammar was used to model the shifts. A multi-cut L-shaped method was used to solve the problem.

Dahmen et al. [37] modeled a multi-activity shift generation and assignment problem with breaks for arbitrary period length. All employees are assumed to be equally capable of carrying out all activities. The model was solved using CPLEX. Akkermans et al. [38] solved a cyclic multi-day shift generation problem with breaks. The solution method used has two phases. In the first phase, an integer linear program is solved to determine the shifts to use with a fuzzy approach applied for breaks. In the second phase, an iterative greedy algorithm is used to schedule the actual breaks into the shifts.

Kletzander & Musliu [39] modeled the Minimum Shift Design (MSD) problem using direct, count-based, and network flow based representations. The resulting models were evaluated on existing benchmark instances using constraint programming and mixed integer programming solvers. The network flow representation was found to be superior to all existing formulations. Shuqing et al. [40] presented a non-cyclic multi-day shift generation problem with hierarchical employee levels. Junior employees are the cheapest but cannot work on their own, and senior employees can only lead a given number of junior employees at any time. A two-stage algorithm was presented. In phase one, the shifts were generated without any consideration for the employee levels. In phase two, the shifts from phase one were given employee levels. A genetic algorithm with improvement heuristics was used to solve the phases iteratively.

Smet et al. [41] proposed a cyclical one-day multi-skill shift generation model with demand smoothing. The idea is to treat extreme peaks in demand by redistributing the demand from the peak to the surrounding timeslots. The maximum number of different shifts and the ratio of different short shifts to all shifts are also limited to offer improved control over extreme peak treatment. Dahmen et al. [42] presented a shift scheduling problem with multiple departments for a period of multiple days. The initial problem was first reduced by aggregating adjacent timeslots into larger timeslots and solving. This reduced problem was then solved using a subset of all shifts that is grown using a column generation based heuristic. A subset of shifts for the original problem was generated based on the solution of the reduced problem, and a variety of heuristics based on decomposition w.r.t. departments and/or time was developed to yield a solution to the original problem using these shifts.

The first major contribution of the task-based shift generation problem was the study by Dowling et al. [18]. They developed a day-to-day planning tool for determining minimal ground staff at an international airport. Their two-stage approach employs a simulated annealing algorithm. First worker shifts are decided over a two-week planning horizon. Then tasks are allocated to shifts/workers. Valls et al. [43] introduced a model to minimize the number of workers performing a machine load plan with unlimited heterogeneous workforce. The vertex coloring based model is solved with a branch-and-bound algorithm.

Krishnamoorthy and Ernst [44] introduced a class of problems called *Personnel Task Scheduling Problems* (*PTSP*). Given the rostered staff on a particular day, the PTSP is to assign tasks with specified start and end times and required skills to available staff. Later, Krishnamoorthy et al. [2] introduced a special case called *Shift Minimization Personnel Task Scheduling Problem* (*SMPTSP*). The objective of the SMPTSP is to minimize the number of employees used to perform the tasks. The problem was called *license and shift class design problem* by Jansen [45] and *tactical fixed interval scheduling problem* by Kroon et al. [46]. The latter also showed that the SMPTSP is NP-hard.

Prot et al. introduced an equity objective to the PTSP [20]. The goal is to find fairer schedules by balancing the employees' workloads. The constraint that all tasks must be assigned is relaxed, yielding the primary objective of minimizing the number of unassigned tasks. The secondary objective is minimizing inequity among the employees.

2.2 Staff rostering

Staff rostering can include the scheduling of days-off and vacations. The working and resting times are constrained by the collective labor agreements and government regulations. The main challenge is to consider the performance of staff on financial efficiency as well as the health, safety, and well-being of the employees. The basic requirements are having the correct number of employees with appropriate competences working at all times (employer perspective) while respecting the employees' working time limits and other contractual obligations and softer well-being targets (employee perspective). A good review of staff rostering can be found in Ernst et al. [4] and. A more recent literature review can be found in [9].

Garey and Johnson [11] and Bartholdi [47] proved the staff rostering problem is NP-hard. Academic-based commercial software packages for staff rostering began to rise in the mid-2000s, see e.g. Burke et al. [48], Bard and Purnomo [49], Beddoe et al. [50], and Bilgin et al. [51]. Staff rostering algorithms have advanced convincingly since then. A multitude of objectives can be simultaneously optimized, and good solutions can be reached in practical running times with reasonable hardware. For examples, see Burke and Curtois [52], Nurmi et al. [53], Jin et al. [54], Gärtner et al. [55], and Kingston [56].

National working hours acts and employment contract acts set regulations for shift work. Additional practices may be agreed at the company level. Human aspects should be considered as well as legal aspects (see Fig. 2). The most important optimization targets are resting times, working hours, work and rest during weekends, and various domain and company specific constraints on work content. For example, in the Finnish public health sector the distribution of night shifts tends to be extremely sensitive.

The susceptibility to health issues and pressure on social life caused by shift work varies between employees. The employees' ability to affect their working hours is crucial to tolerating and even overcoming such effects, decreasing sickness absences, increasing working capacity, and lengthening careers. Personal work time control has been shown to positively influence mental health outcomes such as affective well-being and perceived stress [57]. The positive effects have been ascribed to the improvement of balance between effort and recovery, and between work and non-work life.

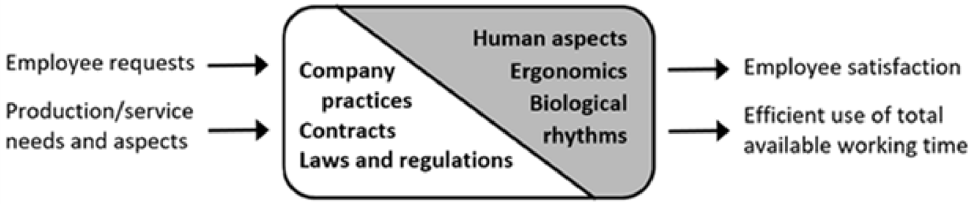

Figure 2. Legal and human aspects of shift work.

In Finland, The Finnish Institute of Occupational Health (FIOH) has extensively studied the effects and consequences of shift work. The institute has published their latest recommendations for shift work [58] in May 2019 based on a ten-year longitudinal study between 2008 and 2018 on 13 000 hospital workers (see e.g. [59] and [60]). Fig. 3 shows the fourteen quantitative risk and stress factors presented by FIOH with a three-week planning horizon.

Recommendations	**Overload →** *Importance ↓*	**Heavy (red)**	**Overload (orange)**	**Increased (yellow)**	**Safe (green)**
1. Length of the workday					
A. Longest work period[1]	*1*	>55h	>48h	>40h	≤40h
B. Working hours in longest shift[2]	*2*	>14h	>12h	>10h	≤10h
C. Consecutive working days[3]	*3*	≥8	7	6 or 2	3-5 (all)
D. Single working days[3]	*3*	≥1			0
2. Timing of the working hours					
A. Consecutive evening shifts[3]	*3*	≥6	5	4	0-3 (all)
B. Consecutive night shifts	*2*	≥6	5	3-4	0 or 2 (all)
C. Single night shifts	*2*				≥0
D. Number of night shifts	*1*	≥9	5-8	3-4	0-2 (all)
3. Recovery and rest times					
A. Rest times below 11h[3]	*1*	≥9	5-8	2-4	0-1 (all)
B. Rest time after a night shift	*3*	<11h	<28h	≤48h	>48h
C. Longest rest period[4]	*3*	<24h	<35h	<48h	≥48h
4. Reconciling work and family life					
A. Free weekends[5]	*2*		0	1	2-3
B. Single days-off	*2*	≥4	3	2	0-1
5. Possibilities to influence personal working times					
A. Employees can make requests	*2*	No			Yes

Figure 3. Fourteen quantitative risk and stress factors and the recommendations for a planning horizon of three weeks by FIOH [58]. Smaller number denotes higher importance.
[1]between two days-off, [2]shift length should not be under 4h, [3]at least one such case, [4]between two shifts, [5]both Saturday and Sunday free

The recommended values for the individual factors are well justified. However, in a recent study [61] it was shown that simultaneously satisfying them all is problematic in real-world scenarios. For example, employee requests can be in direct conflict with e.g. the law, personal work contracts, or the ideal recommendations. Nevertheless, the FIOH recommendations should act as a solid starting point for general staff rostering rules and practices.

2.3 Effects of shift generation on staff rostering

The generated shifts are used as input in staff rostering, where shifts are assigned to the employees (see Fig. 1). Streamlining shift generation and staff rostering ultimately enables an organization's employees to perform at their best.

Next the effect of shift structures on staff rostering optimization is studied. This is the first time such analysis is conducted. The study shows that the generated shift structures have a significant effect on optimizing the staff rosters. The study concerns a real-world instance from a contact center. The solution methods described in Chapter 5 are used to generate optimized shifts and optimized staff rosters.

The *shift generation problem* is given as follows using the model in [I] and [II]:

C1 The sum of working time in the generated shifts must match the sum of the demand for labor as closely as possible.

C2 Excess in shifts (over demand) is minimized.

C3 Shortage in shifts (under demand) is not allowed.

V1 The number of different shifts is minimized.

V3 Shifts of less than 9 hours and over 12 hours (8 and 12 hours for Friday) are minimized.

V4 The target for average shift length is 10 hours.

P1 Shifts cannot start between 22:01 and 06:59.

P2 Night shifts must end before 08:01.

The *staff rostering problem* is given as follows using the model presented in [62]. The timeframe is three weeks.

C1 An employee's shifts must not overlap.

C2 All the generated shifts must be allocated.

R1 An employee's mandatory limits for working hours and working/off days must be respected.

R3 At least one free weekend per employee is guaranteed.

- R5 Eleven hours of rest time is guaranteed between adjacent shifts of an employee.
- R5b 48 hours of rest time after a night shift is guaranteed.
- R7 Employees work consecutively at most four days.
- R7b Employees work consecutively at most four evening shifts.
- R10 A work period between adjacent days-off should be at most 40 hours.
- R11 An employee should have at least one rest period of at least 35 hours each week.
- O1 An employee must have all the competences required for their shifts.
- O5 An employee cannot carry out particular shift combinations on adjacent days.
- O8 Total working time for each employee should be within ±4 hours from their personal target.
- E1 Single days-off should be minimized.
- E2 Single working days should be minimized.
- E8 An employee cannot carry out particular shifts immediately after absent days.
- E10 An employee's maximum number of night shifts is four.
- E11 Single night shifts are not allowed.
- P2 Assign a shift to an employee on a day with a requested day-on. Assign no shift to an employee on a day with a requested day-off.
- P3 Assign a requested shift to an employee on a day with a shift request. Don't assign a requested shift to an employee on a day with a request to avoid a shift.

Three different shift structures were generated for the contact center based on the length of the shifts and the average shift length (see Fig. 4). The allowed shift lengths used vary from strict 8-hour shifts to 6-12-hour shifts, while the target for average shift length varies from 8 hours to as long as possible. The total number of required hours to cover the demand varies from 10617 to 12456 and the number of shifts from 978 to 1557. The average shift length varies between 480 and 651 minutes. Notably, the number of evening shifts (shift structure B) is greatly reduced when longer shifts are preferred (shift structure C). Fig. 5 shows an example how well the optimized shift structure covers the predicted workload.

SHIFT STRUCTURE	A	B	C
Main targets			
C2 and C3 (optimized shift coverage)	Working time in shifts must match demand		
V3 (min and max shift length)	8 – 8	6 – 12	6 - 12
V4 (average shift length)	8	10	max
P1 (shift start time)	Shifts cannot start between 22:01 and 06:59		
P2 (shift end time)	Night shifts must end before 08:01		
Characteristics of the generated shifts			
Total number of required hours	12456	10863	10617
Average shift length	480	525	651
Number of shifts	1557	1242	978
Nbr of morning, evening and night shifts	597, 570, 390	453, 399, 390	453, 142, 383
STAFF ROSTERING SOLUTION	**A**	**B**	**C**
Number of employees needed	149	125	113
Number of hard violations	55	36	19
Number of soft violations	556	116	47
0. Total working time compared to target time			
A. Within ±4 hours (S1)	200	7	1
1. Length of the workday			
A. Longest work period ≤ 40h (S3)	0	1	5
C. Consecutive working days 2-4 (H)	0	0	0
D. Single working days minimized (S1)	143	31	27
2. Timing of the working hours			
A. Consecutive evening shifts 0-4 (S1)	0	0	0
B. Consecutive night shifts 0 or 2-4 (S3)	12	10	1
C. Single night shifts not allowed (H)	0	0	0
D. Number of night shifts 0 or 2-4 (S3)	37	1	0
3. Recovery and rest times			
A. Rest times below 11h 0-1 (S1)	0	1	0
B. Rest time after a night shift ≥ 48h (S3)	10	6	0
C. Longest rest period ≥ 35h (H)	0	0	0
4. Reconciling work and family life			
A. Free weekends 1-3 (H)	55	36	19
B. Single days-off minimized (S1)	164	59	13

Figure 4. The effects of different shift structures on staff rosters.

Next, the staff was rostered using these three different shift structures. The number of employees needed to roster all shifts varies between 113 (longest shifts) and 149 (shortest shifts). Note that this study does not include the employees' requests due to lack of data and the practical difficulty in gathering it for the different shift structures.

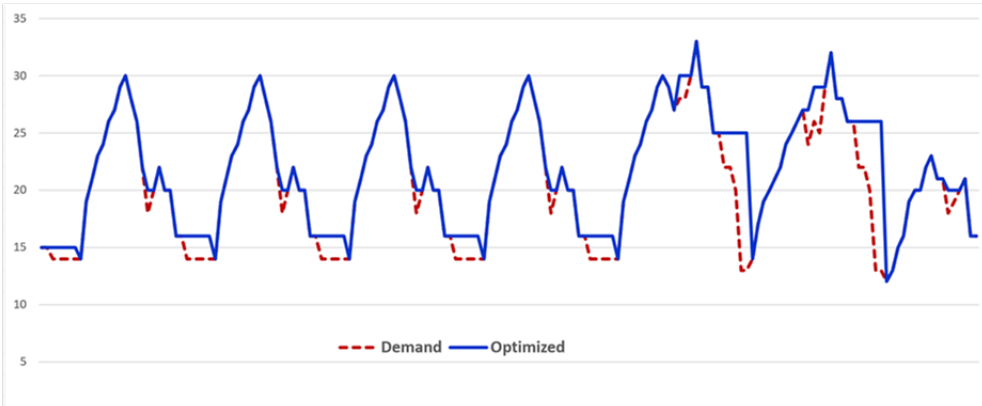

Figure 5. The shift structure covering a one-week demand using scenario B (see Fig. 4).

Fig. 4 shows the effects of the three different shift structures on the possibility to generate acceptable staff rosters. The four most evident observations and findings are the following:

1. Less employees are needed when the average shift length increases.
2. The demand is significantly easier to cover with longer shifts.
3. The stress and risk factors and the FIOH recommendations are better considered with longer shifts.
4. Less evening shifts are needed with longer shifts.
5. Shift structure has no effect on the number of night shifts.

As much as 32% more employees are needed to carry out the schedule when comparing scenarios A and C. This is due to the 17% increase in the total number of required hours to cover the demand, and also partly due to the 60% increase in number of shifts. This, of course, would increase the payload. Even if it could be possible to hire more employees, adopting eight-hour shifts would increase the stress and risk factors to heavy and overload.

On the other hand, versatile shift lengths would enable employees to better target their shift requests, i.e. reconciling their work and family life. For example, an employee might want to have a shorter shift either in the morning or in the evening.

However, Fig. 4 shows that with longer shifts the optimized rosters include more free weekends and less single days-offs.

According to FIOH, the ideal maximum shift length is ten hours. Fig. 4 shows that when the average shift length is ten hours (shift structure B), acceptable staff rosters can be generated. However, 11% more employees are still needed compared to shift structure C. Furthermore, the number of soft violations is increased to 250%.

An important observation not evident from the computational results is the effect of having fewer evening shifts. The adequate rest times between the shifts is then easier to realize. On the other hand, night shifts are the least preferred among the employees. The presented analysis does not provide a solution to this challenge.

Even though the presented study concerns a single instance from a contact center, we believe that the results have wide applicability. The study showed that the generated shift structures have a significant effect on optimizing the staff rosters. The study also showed that we should allow longer working days, because they imply better consideration of the stress and risk factors introduced by FIOH.

3 Employee-Based Shift Generation

Planning horizon is the period of time over which the shifts are generated. The planning horizon is divided into n consecutive, equal-sized *timeslots*. For example, if a single day (00:00 - 24:00) is divided into 24 timeslots, the first timeslot is [00:00, 01:00] and the last timeslot is [23:00, 24:00]. A number of employees for a given *activity* is required at each timeslot. *Demand* is the set of employee requirements over all the timeslots and activities of the planning horizon. For example, a little grocery store might have two activities, namely working as cashier and shelving products. A single cashier is needed at all times, another cashier is needed during rush hours (16:00-19:00), and some shelving work needs to be done in the morning (6:00-10:00).

A *shift* is a mapping of a set of consecutive timeslots to activities. A *solution* is a multiset of shifts. An *active shift* w.r.t. to a solution S is a shift that is used at least once within S. *Excess* at slot s for a given activity a in solution S is the positive difference between the number of times activity a appears in slot s over all the shifts in solution S and the demand in slot s for activity a, and *shortage* is the opposite. For example, Figure 6 shows a scenario where the shifts meet the demand exactly between 00:00 to 07:00 and 16:00 to 00:00. From 07:00 to 08:00 the shifts have in total 10 employees, which yields excess of one employee for an hour since the demand is only 9 employees at the time. Similarly from 08:00 to 16:00 the shifts have in total 20 employees, which yields varying amounts of excess and shortage as the employee demand varies between 12 (08:00 to 09:00) and 30 (12:00 to 13:00) employees.

Given employee demand over time and a set of usable shifts, the *employee-based shift generation problem* is to choose a multiset of shifts s.t. the deviation between employee demand and shift activity is minimized.

3.1 Problem Characteristics

The following characteristics are found in literature. Table 1 lists the less-used options for each characteristic from the relevant articles found in Chapter 1. For example, a model from any article showing on the single-activity row of Table 1 only represents one activity, while a model in an article not appearing on said row can represent multiple activities.

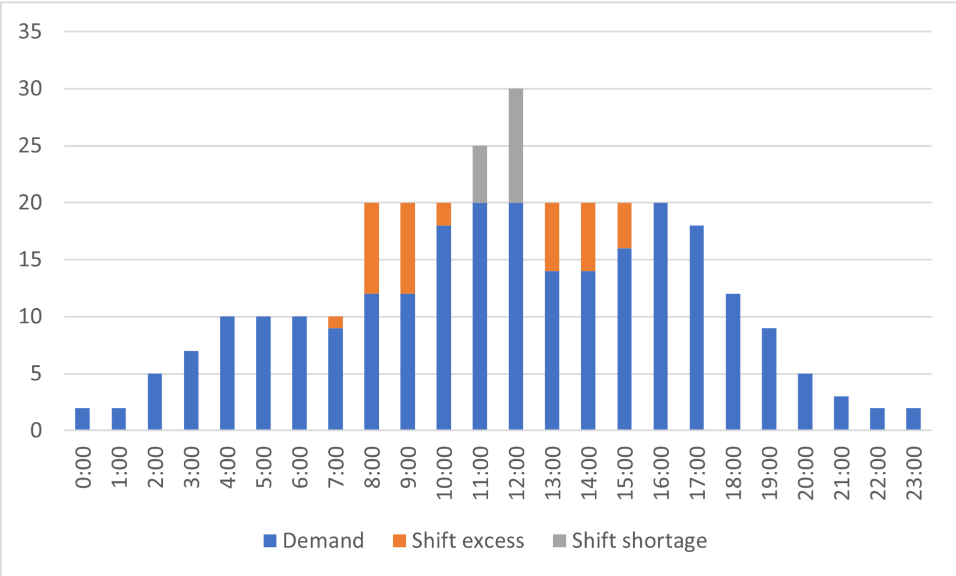

Figure 6. A small instance of the ESG and the deviation from it of a simple feasible solution. Each bar depicts the number of employees.

- The number of different activities can range from one to many. A *single-activity model* is less expressive than a multi-activity model.
- The number of days in a planning horizon can range from one to many. A single-day model should be used in cases where there is no need for night shifts, i.e. the work does not take place 24/7. It is sometimes possible to use *single-day models* for longer planning horizons, but this usually results in worse solutions, as shifts that start and end on different days cannot be fully considered.
- Demand can be *certain* or *uncertain*. Certain demand calls for a static number of employees per timeslot and activity, while with uncertain demand the required number of employees is a random variable or even a decision variable.
- *Shift assignment* means assigning shifts to actual employees to form a *staff roster*. When shift assignment is combined with shift generation, the process is sometimes called *tour scheduling*. In most models shifts are generated without assignment.
- *Breaks* can be scheduled within shifts. This can be especially important in cases with high peaks in demand.
- The *number of different shifts* can be minimized or limited. This can lead to shift structures that are easier to schedule, comprehend and change as more

shifts end up being interchangeable.

- A few models consider *transitions*, i.e. travelling between locations. Travelling takes time, money and can be considered undesirable by employees. Locations may be e.g. different departments in a store or different stores close to each other.
- Larger aggregates of work that should be carried out by a single person can be represented as *tasks*. Very few models consider both demand and tasks.

Table 1. Classification of employee-based shift generation problems.

Single-activity	[7; 8; 23; 24; 25; 28; 30; 32; 34; 38; 39]
Single-day	[24; 27; 32; 41]
Uncertain demand	[30; 34; 41]
Shift assignment	[26; 27; 28; 29; 30; 31; 33; 36; 37; 42]
No breaks	[7; 8; 23; 32; 34; 39; 40; 42]
Shift count minimization/limit	[7; 8; 25; 32; 39; 41]
Transitions	[22; 26; 42]
Tasks	[26; 29]

3.2 A Simple Model

The following model depicts a single-activity shift generation problem without breaks for an arbitrarily long planning horizon. The objective is to minimize weighted excess and shortage. This version of the problem is actually polynomial in the number of usable shifts [8], assuming every shift consists of consecutive activity slots (i.e. the shifts have no breaks). The model has been adapted from [8].

Notation:

T	The set of (consecutive) timeslots.
S	The set of allowed shifts.
d_t	Employee demand at timeslot t.
a_{st}	1 if shift s has activity in slot t, 0 otherwise.
c^+, c^-	Costs for excess and shortage of employees per slot.

Decision variables:

$$
\begin{aligned}
x_s &= \text{the number of copies of shift } s \in S \text{ used.} \\
y_t^+ &= \text{the excess in timeslot } t \in T. \\
y_t^- &= \text{the shortage in timeslot } t \in T.
\end{aligned}
$$

Problem-ESG:

$$Z_{ESG} = min\left(\sum_{t\in T} c^{+}y_t^{+} + \sum_{t\in T} c^{-}y_t^{-}\right) \quad \text{s.t.} \tag{1}$$

$$\sum_{s\in S} a_{st}x_s = d_t + y_t^{+} - y_t^{-} \ \ \forall t \in T \tag{2}$$

$$x_s \geq 0, x_s \in \mathbb{Z} \ \ \forall s \in S \tag{3}$$

$$y_t^{+}, y_t^{-} \geq 0 \ \ \forall t \in T \tag{4}$$

The objective function (1) minimizes the weighted sum of deviations from demand.

Equation 2 links excess and shortage to selected shifts.

Equations 3 and 4 set the domains on the decision variables. Note that y_t^{+}, y_t^{-} are always integral.

4 Task-Based Shift Generation

A *task* is a piece of work that takes some time (*duration*), has to be carried out somewhere (*location*) at some time (within a *time window*, i.e. the task needs to start between its earliest and latest starting time), and may require some competence, permit etc. (*skills*) of the person carrying it out. An employee may use some *mode of transport* to move (*transition*) between locations. The time taken by traveling (*transition time*) that varies between different locations must also be considered. A *precedence constraint* between two tasks either enforces one to precede the other (i.e. task t_1 must end before task t_2 begins), or both to occur simultaneously (i.e. task t_1 must start at the same time as task t_2).

The problem is to assign a set of tasks of arbitrary length to a set of possible shifts. The tasks may contain e.g. skills or precedence constraints (task t_1 must be carried out before/during/after task t_2) that must be considered. The tasks may have unique locations s.t. transition times between tasks must be considered.

An example of a task-based shift generation problem is given in Fig. 7. The rectangles indicate the strict times when the tasks must be carried out. The letters indicate employees able to carry out a task. The parentheses and the colors indicate assignment of tasks to employees and shifts, respectively, of one possible solution. The solution contains 6 shifts, which is the smallest possible number as 6 tasks overlap at 11 o'clock. There are in total 18 feasible (shift, employee) pairs, i.e. the 6 shifts can be assigned to the employees in total 18 times if each shift and employee is considered in isolation.

4.1 SMPTSP

Given tasks and employees, the Shift Minimization Personnel Task Scheduling Problem (SMPTSP) is to assign all the tasks to the minimum number of employees s.t. the employees have the requisites to carry out the tasks assigned to them, and that no two tasks assigned to the same employee overlap in time. The problem was first introduced and modelled by Krishnamoorthy et al. [2] as a specific type of Personnel Task Scheduling Problem [44]. The problem is NP-hard [46].

An example of SMPTSP is given in Fig. 7. The rectangles indicate the strict times when the tasks must be carried out. The letters indicate employees able to carry out a task. The parentheses and the colors indicate assignment of tasks to

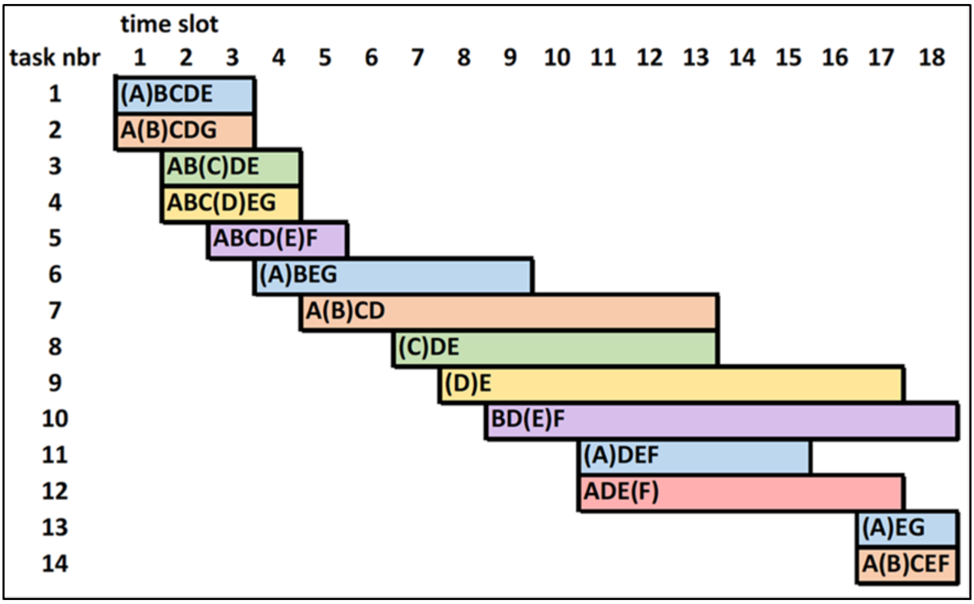

Figure 7. A small task-based instance without transitions, along with a feasible solution, as presented by Nurmi et al. [V].

employees and shifts, respectively, of one possible solution. The solution contains 6 shifts, which is the smallest possible number as 6 tasks overlap at 11 o'clock.

This NP-hard problem can be formulated as follows [2]. Notation:

J	The set of tasks. Each task has a starting time, ending time and a set of skills required to carry it out.
P	The set of employees.
$P_j \subseteq P$	The set of employees that may carry out task j.
K_t	The set of tasks active at time t.
K_t^w	The set of tasks active at time t that worker w can perform.
C	The family of sets containing all such sets K_t that $K_t \not\subseteq K_{t'}\ \forall t \neq t'$.
C^w	The family of sets containing all such sets K_t^w that $K_t^w \not\subseteq K_{t'}^w\ \forall t \neq t'$.

Decision variables:

$$x_{jw} = \begin{cases} 1 & \text{if task } j \in J \text{ is assigned to employee } w \in W \\ 0 & \text{otherwise.} \end{cases}$$

$$y_w = \begin{cases} 1 & \text{if employee } w \in W \text{ is active, i.e. has assigned tasks} \\ 0 & \text{otherwise.} \end{cases}$$

Problem-SMPTSP:

$$Z_{SMPTSP} = min \sum_{w \in W} y_w \tag{5}$$

$$\text{s.t.} \sum_{w \in P_j} x_{jw} = 1 \quad \forall j \in J \tag{6}$$

$$\sum_{j \in K_t^w} x_{jw} \leq y_w \quad \forall w \in W, K_t^w \in C_w \tag{7}$$

$$x_{jw} \in \{0, 1\} \quad \forall j \in J, w \in W \tag{8}$$

$$0 \leq y_w \leq 1 \quad \forall w \in W \tag{9}$$

The objective function (5) minimizes the number of used employees.

Equation 6 ensures that each task will be carried out by exactly one able worker.

Equation 7 ensures that no overlapping tasks are assigned to any employee, and that the indicator for using a worker actually indicates worker usage.

Equations 8 and 9 set the bounds on the decision variables. Note that y_w will only get integral values due to equations 5 and 7.

4.2 ESMPTSP

Given tasks and employees, the Extended Shift Minimization Personnel Task Scheduling Problem (ESMPTSP) is to assign all the tasks to the minimum number of employees s.t. the employees have the requisites to carry out the tasks assigned to them, and that no two tasks assigned to the same employee overlap in time. Additionally, shifts (i.e. all the task assignments of a single employee) are rewarded for each additional employee that can carry out the same shift - the number of feasible (shift, employee) pairs is maximized. This leads to improved interchangeability of shifts between employees and leads to e.g. easier management of ad-hoc absences. The problem was first introduced, along with a mathematical model, by Kyngäs et al. [VI] as an extension to the SMPTSP [2].

An example of ESMPTSP is given in Fig. 7. The rectangles indicate the strict times when the tasks must be carried out. The letters indicate employees able to carry out a task. The parentheses and the colors indicate assignment of tasks to employees and shifts, respectively, of one possible solution. The solution contains 6 shifts, which is the smallest possible number as 6 tasks overlap at 11 o'clock. There are in total 18 feasible (shift, employee) pairs, i.e. the 6 shifts can be assigned to the employees in total 18 times if each shift and employee is considered in isolation.

This NP-hard problem can be formulated as follows [VI]. Notation:

J	The set of tasks. Each task has a starting time, ending time and a set of skills required to carry it out.
P	The set of employees.
$P_j \subseteq P$	The set of employees that may carry out task j.
K_t	The set of tasks active at time t.
K_t^w	The set of tasks active at time t that worker w can perform.
C	The family of sets containing all such sets K_t that $K_t \nsubseteq K_{t'}\ \forall t \neq t'$.
C^w	The family of sets containing all such sets K_t^w that $K_t^w \nsubseteq K_{t'}^w\ \forall t \neq t'$.
α	The weight for the number of shifts in the objective function.
β	The weight for the number of feasible (shift, employee) pairs in the objective function.

Decision variables:

$$x_{jwv} = \begin{cases} 1 & \text{if task } j \in J \text{ is assigned to employee } w \in W \text{ and } v \in P_j \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{wv} = \begin{cases} 1 & \text{if employee } w \in W \text{ is active and employee } v \in W \text{ can} \\ & \text{carry out the shift of } w \\ 0 & \text{otherwise.} \end{cases}$$

For $w \neq v$, call x_{jwv} pseudoassignments of v to w with respect to j, as they represent whether task j could be assigned to v assuming j is assigned to w. Similarly we call y_{wv} pseudoassignments of v to w, as they represent whether all the tasks (and thus the entire shift) assigned to w could be assigned to v.

Problem-ESMPTSP:

$$Z_{ESMPTSP} = min\left(\alpha * \sum_{w \in W} y_{ww} - \beta * \sum_{w,v \in W} y_{wv}\right) \tag{10}$$

$$\text{s.t.} \sum_{w \in P_j} x_{jww} = 1 \quad \forall j \in J \tag{11}$$

$$\sum_{j \in K_t^w} x_{jww} \leq y_{ww} \quad \forall w \in W, K_t^w \in C_w \tag{12}$$

$$y_{wv} \leq x_{jwv} - x_{jww} + 1 \quad \forall (j,w,v) \in J \times W \times W : w \neq v \tag{13}$$

$$y_{wv} \leq \sum_{j \in J} x_{jww} \quad \forall (w,v) \in W \times W \tag{14}$$

$$x_{jwv} = 0 \quad \forall (j,w,v) \in J \times W \times W : w \notin P_j \text{ or } v \notin P_j \tag{15}$$

$$x_{jwv} = x_{jww} \quad \forall (j,w,v) \in J \times W \times W : w,v \in P_j \tag{16}$$

$$x_{jwv} \in \{0,1\} \quad \forall (j,w,v) \in J \times W \times W \tag{17}$$

$$y_{wv} \in \{0,1\} \quad \forall w \in W, v \in W \tag{18}$$

The objective function (equation 10) is a weighted sum of the number of used employees and the number of able (employee, shift) pairs.

Equation 11 ensures that each task will be carried out by exactly one able worker.

Equation 12 ensures that at most one task per clique (maximal set of overlapping tasks) is assigned to a single employee, and that the indicator for using a worker indicates worker usage.

Equation 13 ensures that a shift cannot be pseudoassigned to a worker if it has tasks the worker is unable to do.

Equation 14 ensures that empty shifts are not counted as pseudoassignments.

Equations 15 and 16 ensure tasks are pseudoassigned according to both actual assignments and the abilities of the workers.

Equations 17 and 18 force the variables to be binary.

4.3 GTSGP

The General Task-based Shift Generation Problem (GTSGP) was first introduced by Nurmi et al. [III] as a task-based shift generation problem. The mathematical model was presented by Kyngäs et al. [IV]. In addition to the components present in ESMPTSP, the GTSGP models time windows for tasks, travel between tasks, and shift-local precedence constraints between tasks.

An example of GTSGP is given in Fig. 7. The rectangles indicate the strict times when the tasks must be carried out. The letters indicate employees able to carry out a task. The parentheses and the colors indicate assignment of tasks to employees and shifts, respectively, of one possible solution. The solution contains 6 shifts, which is the smallest possible number as 6 tasks overlap at 11 o'clock. There are in total 18 feasible (shift, employee) pairs, i.e. the 6 shifts can be assigned to the employees in total 18 times if each shift and employee is considered in isolation.

This NP-hard problem can be formulated as follows [IV]. Notation:

- S The resulting set of shifts.
- TS The planning horizon, i.e. the contiguous set of timeslots.
- T The set of tasks.

E	The set of employees.
L	The set of locations.
C	The set of skills.
R	The set of transport types.
G	The set of task types.
M	The travel times of possible transitions. For any given transport type, the travel times must adhere to the triangle inequality, yet multiple modes of transport may be used.
P	The set of shift-local precedence constraints.
c_e	The cost of using employee e in a single shift.
elb_e	Lower bound (incl.) on total working time of employee $e \in E$.
eub_e	Upper bound (incl.) on total working time of employee $e \in E$.
l_e^s	Starting location of employee $e \in E$.
l_e^e	Ending location of employee $e \in E$.
r_e	Transport type of employee $e \in E$.
a_e^s	Earliest timeslot (incl.) of availability of employee $e \in E$.
l_e^s	Latest timeslot (incl.) of availability of employee $e \in E$.
d_t	Duration of task $t \in T$.
tlb_t	Earliest timeslot (incl.) for starting time of task $t \in T$.
tub_t	Latest timeslot (incl.) for ending time of task $t \in T$.
g_t	Task type of task $t \in T$.
l_t	Location of task $t \in T$.
ttc_r	Task transition working time coefficient of transport type $r \in R$ ($ttc_r \in [0, 1]$). Transition time between two tasks is considered working time with ttc_r as coefficient, i.e. working time = transition time * ttc_r.
etc_r	Employee transition working time coefficient of transport type $r \in R$ ($etc_r \in [0, 1]$). Transition time to/from the first/last task of a shift is considered working time with etc_r as coefficient, i.e. working time = transition time * etc_r.
eta_r	Employee transition availability requirement of transport type $r \in R$ ($eta_r \in \{0, 1\}$), i.e. must the first/last transition of a shift be considered with respect to the assigned employee's availability?
$m_{rll'}$	Travel time from location $l \in L$ to location $l' \in L$ using transport type $r \in R$ ($m_{rll'} \in M$).

$p_{g,g'}$ Shift-local precedence constraint that dictates tasks of task type $g \in G$ cannot be succeeded by tasks of task type $g' \in G$.

q_{et} Does employee e have the skills necessary to do task t? ($q_{et} \in \{0, 1\}$).

T_e The set of tasks doable by employee $e \in E$ ($T_e \subseteq T$).

E_t The set of employees capable of carrying out task $t \in T$ ($E_t \subseteq E$).

α The weight of the cost of employees used in the objective function.

β The weight of feasible (shift, employee) pairs in the objective function.

γ The weight of traveling time in the objective function.

Decision variables:

$$x^s_{et} = \begin{cases} 1 & \text{if for shift } s(e) \text{ the first task is } t, \\ 0 & \text{otherwise.} \end{cases}$$

$$x_{ett'} = \begin{cases} 1 & \text{if task } t \text{ is immediately followed by task } t' \text{ in shift } s(e) \text{ with } t \neq t' \\ 0 & \text{otherwise.} \end{cases}$$

$$x^f_{et} = \begin{cases} 1 & \text{if for shift } s(e) \text{ the last task is } t, \\ 0 & \text{otherwise.} \end{cases}$$

$$b_{et} = \begin{cases} 1 & \text{if task } t \text{ belongs to shift } s(e) \\ 0 & \text{otherwise.} \end{cases}$$

$$sx_e = \begin{cases} 1 & \text{if shift } s(e) \text{ is non-empty} \\ 0 & \text{otherwise.} \end{cases}$$

$y_{et} =$ starting time (i.e. first slot) of task $t \in T$ assuming the shift it belongs to is assigned to employee $e \in E$.

$wt_{ee'} =$ working time for employee $e \in E$ assuming e carries out the shift assigned to employee $e' \in E$.

$$ve_{e'e} = \begin{cases} 1 & \text{if employee } e \text{ may not carry out shift } s(e') \\ 0 & \text{otherwise.} \end{cases}$$

$$vs_{e'e} = \begin{cases} \geq 1 & \text{if employee } e \text{ does not have the skills for shift } s(e') \\ 0 & \text{otherwise.} \end{cases}$$

$$vwu_{e'e} = \begin{cases} \geq 1 & \text{if shift } s(e') \text{ has too much working time for employee } e \\ 0 & \text{otherwise.} \end{cases}$$

$$vwl_{e'e} = \begin{cases} \geq 1 & \text{if shift } s(e') \text{ has too little working time for employee } e \\ 0 & \text{otherwise.} \end{cases}$$

$$vtu_{e'et} = \begin{cases} \geq 1 & \text{if employee } e \text{ would be late at task } t \text{ if carrying out shift } s(e') \\ 0 & \text{otherwise.} \end{cases}$$

$$vteu_{e'et} = \begin{cases} \geq 1 & \text{if employee } e \text{ would be late at their end location due to} \\ & \text{the last task } t \text{ of shift } s(e') \text{ if carrying out shift } s(e') \\ 0 & \text{otherwise.} \end{cases}$$

Problem-GTSGP:

$$Z_{GTSGP} = min\Bigg(\alpha * \sum_{e} sx_e c_e \tag{19}$$

$$- \beta * \sum_{e,e'} (1 - ve_{e'e}) \tag{20}$$

$$+ \gamma * \Big(\sum_{e} wt_{ee} - \sum_{t} d_t \Big) \Bigg) \tag{21}$$

s.t.

$$\sum_{t} x^s_{et} \leq 1 \;\; \forall e \in E \tag{22}$$

$$\sum_{t} x^f_{et} \leq 1 \;\; \forall e \in E \tag{23}$$

$$\sum_{e} b_{et} = 1 \;\; \forall t \in T \tag{24}$$

$$b_{et} = x^s_{et} + \sum_{t'} x_{et't} \;\; \forall e \in E, t \in T \tag{25}$$

$$b_{et} = \sum_{t'} x_{ett'} + x^f_{et} \;\; \forall e \in E, t \in T \tag{26}$$

$$sx_e \leq \sum_{t} b_{et} \;\; \forall e \in E \tag{27}$$

$$sx_e \geq b_{et} \;\; \forall e \in E, t \in T \tag{28}$$

$$y_{et} \geq tlb_t \;\; \forall e \in E, t \in T \tag{29}$$

$$y_{et} \geq a^s_e + eta_{r_e} m_{r_e l^s_e l_t} \;\; \forall e \in E, t \in T \tag{30}$$

$$y_{et'} + d_{t'} + m_{r_e l_{t'} l_t} \leq y_{et} + M_1(1 - \sum_{e'} x_{e't't})$$

$$\forall e \in E, t, t' \in T \tag{31}$$

$$y_{et'} + d_{t'} + m_{r_e l_{t'} l_t} \leq y_{et} + (2 - b_{et} - b_{et'}) * M_1$$

$$\forall e \in E, t, t' \in T : p_{g_t, g_{t'}} \in P \tag{32}$$

$$wt_{ee'} = etc_{r_e} \sum_{t} \Big(m_{r_e l^s_e l_t} x^s_{e't} + m_{r_e l_t l^e_e} x^f_{e't} \Big)$$

$$+\sum_t \left(ttc_{r_e} \sum_{t'} m_{r_e l_t l_{t'}} x_{e'tt'} \right)$$
$$+\sum_t d_t b_{e't} \;\; \forall e \in E, e' \in E \tag{33}$$
$$ve_{ee} = 0 \;\; \forall e \in E \tag{34}$$
$$vs_{e'e} \geq b_{e't}(1 - q_{et}) \;\; \forall t \in T, \forall e, e' \in E \tag{35}$$
$$vwu_{e'e} \geq wt_{ee'} - eub_e \;\; \forall e, e' \in E \tag{36}$$
$$vwl_{e'e} \geq elb_e - wt_{ee'} \;\; \forall e, e' \in E \tag{37}$$
$$vtu_{e'et} \geq y_{et} - tub_t + d_t - 1 + M_1 * (b_{e't} - 1) \quad \forall t \in T, \forall e, e' \in E \tag{38}$$
$$vteu_{e'et} \geq y_{et} + d_t + eta_{r_e} m_{r_e l_t l_e^e} - a_e^s - 1 - M_1 * (1 - x_{e't}^f) \quad \forall t \in T, \forall e, e' \in E \tag{39}$$
$$M_2 * ve_{e'e} \geq vs_{e'e} + vwu_{e'e} + vwl_{e'e} + \sum_t (vtu_{e'et} + vteu_{e'et}) \quad \forall e, e' \in E \tag{40}$$
$$x_{et}^s = x_{et}^f = 0 \;\; \forall e \in E, t \in T : t \notin T_e \tag{41}$$
$$x_{ett'} = 0 \;\; \forall e \in E, t, t' \in T : e \notin E_t \cap E_{t'} \tag{42}$$
$$x_{et}^s, x_{ett'}, x_{et}^f \in \{0, 1\} \;\; \forall e, e' \in E, t \in T \tag{43}$$
$$b_{et}, sx_e, ve_{e'e} \in \{0, 1\} \;\; \forall e, e' \in E, t \in T \tag{44}$$

The objective function is a weighted sum of minimizing the cost of used employees (19) and traveling time considered working time (21) and maximing so-called able pairs (20), i.e. the number of (shift, employee) pairs where the employee may carry out the shift.

The sequential structure of individual shifts is considered in equations (22) - (26). Each nonempty shift has a first task, a last task, and a possibly empty chain of tasks between them. A single task belongs to exactly one shift. Equations (27) and (28) ensure sx_e indicates nonemptiness of the shift corresponding to employee $e \in E$. Equation (27) is actually redundant from the viewpoint of the model, but it is included for the sake of clarity.

Equations (29) - (31) enforce lower bounds on starting times of tasks. The respected bounds are the earliest starting time of the task, the earliest availability of the employee and preceding task in the shift.

Equation (32) ensures that shift-local precedence constraints are respected whenever constrained tasks are assigned to the same shift.

Equation (33) calculates working times for (shift, employee) pairs.

Equation (34) ensures that each employee may carry out their corresponding shift.

Equations (35) - (39) calculate the violations in individual constraints between all (shift, employee) pairs. Equation (40) composes these into indicator variables to signal whether individual pairs are compatible.

All parameters affecting y_{et} have integral values, hence y_{et} will always be integral.

5 Solution Methods

Efficient solution methods exist for the employee-based shift generation problem, see Table 1. The simple employee-based model presented in Chapter 3 is solvable in polynomial time with respect to the number of different shifts available as a network flow problem, assuming every shift consists solely of consecutive activities (i.e. there are no breaks in any shift) [8]. Adding virtually any relevant component to the model, for example multiple activities, breaks, or assignment of shifts to employees, makes the problem NP-hard, see e.g. [11; 12; 13; 14; 15].

In the Minimum Shift Design problem [8], the number of different shifts over the planning horizon must be minimized along with deviation from the employee demand. Kletzander & Musliu [39] presented an exact network flow based model for the problem that proved computationally superior to other existing models, yielding first proven optimal solutions to well-known benchmark instances with Gurobi and CPLEX.

There are eight notable methods for the SMPTSP, of which three are purely heuristic [2; 63; 64], two are exact [65; 66], and three are hybrid methods [1; 67; 68]. The best currently known solution method for the SMPTSP was given by Chandrasekharan et al. [68], followed by Nurmi and Kyngäs [V]. Chandrasekharan et al. solved to optimality 242/247 of the well-known benchmark instances [69]. Nurmi and Kyngäs selected 47/247 hard instances to solve based on how difficult they had been for other published solution methods, and solved 44/47 to optimality as opposed to 42/47 [68]. It is hard to derive state-of-the-art from these diverse methods.

There are only three papers available on the recently introduced problems of the ESMPTSP [VI] and the GTSGP [III, IV].

5.1 PEASTP

The PEAST algorithm [5] is a population-based local search method that was developed to solve real-world scheduling problems. In this thesis we give PEASTP [V], a parallelized version of PEAST. The PEASTP algorithm combines features from some well-known metaheuristics: genetic algorithms [70], ejection chains [71], tabu search [72], and simulated annealing [73]. The metaheuristic has been used for commercial purposes for several years, for example in staff rostering [74] and in professional sports league scheduling [75; 76]. PEAST and PEASTP have been applied to various shift generation problems [I, II, III, IV, V].

5.2 Solyali Lower Bounding Process for the SMPTSP

Solyali [65] developed a process for finding lower bounds for the minimum number of shifts in SMPTSP instances. The process exploits the idea that the set of workers that are used to perform any given set of tasks S that overlap in time has to contain at least $|S|$ workers. Since there are an exponential number of such sets, it is beneficial to not add constraints for all such sets in the model at once. A logical starting point is having a constraint for each maximal overlapping set. New constraints are dynamically added for violated subsets in a branch-and-cut framework.

In practice, the process yields high quality lower bounds for very little computation time. The process was used to find lower bounds for the minimum number of employees of the new benchmark instances in [VI]. While the instances were generated in such a way that the total number of employees was a good estimate for the minimum, the Solyali process quickly revealed instances that could have lower minimums to be confirmed with other methods.

5.3 New Ruin and Recreate Heuristic

This version of ruin and recreate heuristic (2RH, first presented in [VI]) based on [77] was created to solve practical GTSGP instances. Pseudocode for the 2RH algorithm is given in Figure 8.

5.4 Implementation Notes on the 2RH

The implementation language was chosen based on performance and the ability to try out arbitrary high or low level optimizations. Thus C++ was chosen.

By far the most crucial part of the 2RH is evaluating the cost of a single addition. It is usually beneficial to use more storage resources (main memory, cache space) and more calculation when applying moves if it speeds up evaluation of addition.

The time windows in the problems are by nature hard, i.e. a task's starting time must be within the given constraints. However, they can be implemented as soft constraints in order to allow interim solutions with constraint violations to better traverse the search space. They are the most significant time sink with respect to calculation time. To use the reoptimization data [78] as efficiently as possible, the data has to be stored in adjacent memory. It is interesting to note that no matter how the data was stored, using AVX-256 to do the time window calculations of up to 8 tasks at once was much slower than letting the compiler (Intel Compiler 19.1) optimize the non-vectorized code on the test machine. This was the case even after the time window implementation was changed to treat the constraint as soft (e.g. violations in time windows were allowed in intermediate solutions, as opposed to stopping move evaluation and rejecting the candidate move if a violation was found), which removes the bias caused by the fact that the vectorized version would do spurious work compared to the non-vectorized version.

```
Input: a GTSGP instance including e.g. the set of tasks T and
  the set of employees E; internal algorithm parameters; starting
  solution currentSol
Output: the best solution found
round ← 0, bestSol ← null
while round < f do
    storedSol ← currentSol
    seed ← random currently assigned task from T
    tm ← maximum number of tasks to ruin per shift
    tasks ← list of all tasks ordered by distance from seed
    S = ∅
    for t ∈ tasks
        if S(t) ∉ S
            l ← U (1, min(|S(t)|, tm))
            Remove a random string of l tasks from S(t)
              in currentSol
            Update ruin quota
            S ← S ∪ S(t)
        end if
        if ruin quota is full
            break
        end if
    end for
    tasks ← list of all unassigned tasks in unassignment
            count order
    for t ∈ tasks
        P ← all spots in all current shifts where adding t is
           feasible in worsening order of objective
        for p ∈ P
            if U (1, 100) ≥ lowLevelSkipChance
                Add task t to spot p in currentSol
                break
            end if
        end for
    end for
    Update bestSol if necessary
    if U (1, 100) ≤ highLevelSkipChance
        currentSol ← storedSol
    end if
    round ← round + 1
end while
```

Figure 8. Pseudocode for the 2RH algorithm, as presented by Nurmi et al. [VI].

6 Results

In [I] four real-world industry benchmark instances of an employee-based shift generation problem were introduced. The goal was to stick to the demand while minimizing the number of non-eight-hour-shifts and keeping the average length of the shifts as close to six hours as possible. The instances were solved with the PEAST algorithm. The results showed that shorter timeslots lead to better results and high, sharp peaks in demand lead to shorter shifts.

In [II] five real-world industry benchmark instances of an employee-based shift generation problem covering the period of one week were introduced. The instances contain three different task types with the demand split into 15-minute time slots between 6am and 10pm every day. The number and length of breaks per shift depends on the shift length. The results showed that multiple task types and breaks could be scheduled in the shifts with the PEAST algorithm.

In [III] eight task-based shift generation problems were solved with the PEAST algorithm. Seven of the problems were SMPTSP problems from the KEB dataset [2] and one was a simple GTSGP problem, so few features of the full GTSGP model were used. The number of tasks per instance varied between 40 and 665. It was shown that the PEAST algorithm can be used for this kind of problems, but not very effectively.

In [IV] the GTSGP was formally modelled and 15 new benchmark instances were introduced. The number of tasks per instance varied between 18 and 425. Different aspects of the GTSGP, such as precedence constraints, travel costs, and maximum total worker time, were present in the instances. It was shown that the PEAST algorithm can be used for this kind of problems, but transportation between tasks was not handled particularly well. CPLEX and Gurobi unsurprisingly yielded high quality solutions fast on smaller instances but failed to find solutions for larger instances.

In [V] a set of 47 historically challenging SMPTSP instances were solved with a new hybrid algorithm created for the more general GTSGP problems. The number of tasks per instance varied between 104 and 1577, while the number of resulting shifts varied between 20 and 320. The hybrid algorithm found more optimal solutions than any other solution method had, but not on the SWMB dataset [3].

In [VI] a set of 56 historically challenging SMPTSP instances and 30 new benchmark instances were solved as ESMPTSP problems using a new ruin and recreate heuristic. The number of tasks per instance varied between 59 and 2473, while the number of resulting shifts varied between 20 and 496. The results showed that the ratio of feasible pairs to the number of generated shifts varied greatly between instances, meaning that the interchangeability of the resulting shifts varied as well. At worst the ratio was exactly one, indicating a unique correspondence between the shifts and the employees. At best the ratio was greater than two, indicating that on average each shift could be carried out by two different employees.

7 Discussion

7.1 Research questions

7.1.1 Employee-based demand can be efficiently met with algorithmically generated shifts whose structure is not rigorously constrained

Employee-based models have direct application areas in e.g. hospitals, retail stores, and call centers. The main goal is the performance of staff and financial efficiency. Other important goals include fairer workloads and employee satisfaction.

Unlimited shift generation problems with and without breaks were introduced. This was the first time such problems were introduced. The unconstrainedness of an individual shift's structure provides flexibility to the model that might be hard to achieve otherwise. The usefulness of the model was demonstrated by real-world benchmark instances.

At the same time, the chosen solution method is a generic method capable of solving various NP-hard problems and not particularly suited for shift generation. Therefore good solutions are not generated fast. This can be problematic for real-world adoption in cases where quick prototyping of different shift structures for a client is highly beneficial. [8]

7.1.2 Task-based demand can be efficiently met with algorithmically generated shifts

Task-based models have direct application areas in e.g. home care, cleaning, and delivery of goods. The goals are similar to those of employee-based models as presented in the previous section.

An existing problem from literature, SMPTSP, was considered. The SMPTSP has direct practical applications in task-based settings with negligible transition times with many available on-demand workers.

Two new problems, GTSGP and ESMPTSP, were introduced. The SMPTSP is too simple to handle many practical real-world cases in task-based shift generation. For example, in practical applications tasks must often be completed within a time window instead of being fixed in time. Both GTSGP and ESMPTSP aim to make the resulting shift structure more flexible for staff rostering and resilient to inavoidable daily changes that stem from e.g. sick leaves and changes in requirements. In practice, a solution with an equal number of shifts but more feasible pairs is better because on average each shift has more assignable employees, making staff rostering and daily changes easier.

SMPTSP and ESMPTSP are solved with a combination of PEASTP and 2RH to reach high quality solutions much faster than with PEASTP alone. The SMPTSP solutions are better than any others in the literature. Since 2RH achieves good results by itself, in real applications PEASTP can be left out in order to speed up the solving process considerably. As a more generic method developed for GTSGP, it is still not as fast as existing SMPTSP solutions.

7.1.3 Shift structure optimization has significant effect on the achievable staff rosters

Employee-based and task-based problems can be solved efficiently, as demonstrated in research questions one and two. In Section 2.3, the effect of shift structure on the final rosters was examined for the first time. The experiment was not too comprehensive, including only a single instance from a Finnish contact center. However, it underlines well the significant impact of shift structure on e.g. the number of required employees. In practice shift structure is often highly constrained by legislative or practical human considerations, greatly affecting efficiency. For example, the extreme demand peaks often encountered in airports would require unreasonably short shifts to fulfill without major overstaffing.

In Section 2.3 it was found that increasing the average shift length from 8 to 10 hours yielded significant improvements in both shift structure and staff rosters. In the 4-year registry study by Vedaa et al. [79], long shifts were found to be associated with less sick leave days. A potential explanation offered to this relationship was extra days off and their rejuvenative effects. In the shift work study considering industry workers in large Finnish companies [80], the employees were found to sleep and feel better when the length of the working day was twelve hours instead of eight hours. This demonstrates the potential positive effects of non-standard shift lengths.

7.2 Other considerations

The solution methods used are heuristics made for finding a single best solution. Thus, a human planner using these methods will not be provided a host of good (e.g. Pareto optimal) alternatives to choose from. However, the methods can be modified to provide a set of reasonably good solutions with little computational overhead if necessary.

Most solution methods introduced are made for solution quality over speed. Thus, they might provide relatively low quality results with low computation resources, and more greedy solution methods may provide better results in such settings. Adoption by users might also prove challenging in practice. However, in real-world applications it is often advantageous to use more computation time in order to generate shifts and rosters that are of high value and acceptable considering their release time.

Continuous problems with planning horizons of longer than one day were not considered in the present work. While this might limit real-world applicability in some cases, in many cases it is acceptable. For example, functions that either are not operated around the clock or have near-zero minimum employee needs in early morning hours can usually be treated as separate days.

The methods presented yield demonstrably good practical results. However, practical adoption of complex optimization algorithms to complex optimization problems is no trivial task. The optimization algorithm needs various data to function, e.g. HR data and work data. Before optimization, it may be necessary for a human resource planner to modify said data, especially if the other data modules were not designed with optimization in mind. The optimization algorithm needs hardware to run, the amount of which depends on e.g. the number of customers and the interleavedness of their planning schedules. After optimization has been done, a human planner may once again need to make changes to the plan, or reoptimize it fully or partially. After the plan has been carried out, employees need to be paid.

Without all of these auxiliary functions, the greatest optimization algorithm in the world by itself is useless. There are at least two ways to approach the problem: either attach the algorithm to an existing system, or build a new system designed to leverage optimization as a central function. The former approach can be easier and faster, and lead to getting the work out to real clients quickly. However, the end result might not be satisfactory. A system originally designed solely for manual planning will almost certainly not be able to leverage optimization fully. If the algorithms were developed without knowledge of the target system, incompatibilities between the data required by the optimization and the data provided by the system may rise. Building a new system designed from the ground up to leverage optimization at every level will almost certainly yield a better product, but it is a vastly more complex and expensive undertaking. It is also more risky, as there is no guarantee of getting clients due to e.g. newness, complexity, or price. Meanwhile an existing system usually has existing, convertible clients.

Based on a ten-year experience in cooperating with workforce management software companies and working closely with customers and end-users, in practice the usefulness and utility of the optimized shift structures and rosters depends heavily on

1. willingness and effort to map, reconsider, and renew the workforce scheduling process and practices, from the management to the resource planners down to

the individual employees,

2. high-quality input data that potentially requires additional effort compared to manual planning to maintain,

3. centralization of planning and larger planning units, and

4. fairness and transparency towards the employees.

7.3 Future research

7.3.1 The GFA Algorithm for the SMPTSP

The 2RH algorithm was developed to solve a wide variety of GTSGP instances. To focus solely on the SMPTSP, improvements to the best found SMPTSP results can be made.

The pack problem [44] is the problem of maximizing the contribution of a single employee with respect to a given metric. A single employee and a set of tasks with weights are given, and the objective is to maximize the total weight of the tasks carried out by the employee. This problem, also known as weighted interval scheduling, can be efficiently solved in polynomial time [81].

The GFA (Gulls and Flying Ants) algorithm created specifically for SMPTSP is a metaheuristic that combines a powerful pack-based local search operator with a reinforcement learning weighting scheme for the tasks. In each iteration, a subset of employees with tasks assigned are randomly selected. For each such employee in turn, all assigned tasks are unassigned, and the resulting set of both previously and newly unassigned tasks is offered to every employee with no tasks assigned. The best of all examined (remove an employee's tasks, give an employee tasks) pairs of moves is committed. This is followed by offering all remaining free tasks to each employee with at least one task assigned, until no offering improves the solution. Any task unassigned at this point has its weight dynamically grown. A small elitist pool of solutions, in which the worst current solution is replaced by the best current solution each generation, is used.

Crucial components for getting good results include a pool of solutions, using task duration for initial task weights, and increasing task weight for 'difficult' tasks. The heuristic produces results of high quality in well-known benchmark instances reasonably fast. The method presented in [V] produced the best known results on the well-known SMPTSP instances by feeding the 2RH results to the PEASTP algorithm. The GFA alone produces even better results and much faster. GFA produces equal or better results on all the 47 instances solved in [V].

This is ongoing joint research with the developer of the constructive matheuristic [68] that produced the best results on the SMPTSP until this thesis.

7.3.2 The effects of shift generation on staff rosters

In Section 2.3, the effect of shift structure on the final rosters was examined for the first time. Shift structure optimization was found to have a significant effect on the staff rosters. An important finding was that longer working days should be allowed, because they imply better consideration of the stress and risk factors introduced by the Finnish Institute of Occupational Health. PEASTP will be used to justify the findings in the ongoing research. Results obtained earlier in a Finnish contact center will be expanded to other domains.

List of References

[1] Jean-Guillaume Fages and Tanguy Lapègue. Filtering AtMostNValue with difference constraints: Application to the shift minimisation personnel task scheduling problem. *Artificial Intelligence*, 212:116–133, July 2014. ISSN 00043702. doi: 10.1016/j.artint.2014.04.001. URL https://linkinghub.elsevier.com/retrieve/pii/S0004370214000423.

[2] M. Krishnamoorthy, A.T. Ernst, and D. Baatar. Algorithms for large scale Shift Minimisation Personnel Task Scheduling Problems. *European Journal of Operational Research*, 219(1):34–48, May 2012. ISSN 03772217. doi: 10.1016/j.ejor.2011.11.034. URL https://linkinghub.elsevier.com/retrieve/pii/S0377221711010435.

[3] Pieter Smet, Tony Wauters, Mihail Mihaylov, and Greet Vanden Berghe. The shift minimisation personnel task scheduling problem: A new hybrid approach and computational insights. *Omega*, 46:64–73, July 2014. ISSN 03050483. doi: 10.1016/j.omega.2014.02.003. URL https://linkinghub.elsevier.com/retrieve/pii/S0305048314000176.

[4] A.T Ernst, H Jiang, M Krishnamoorthy, and D Sier. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27, February 2004. ISSN 03772217. doi: 10.1016/S0377-2217(03)00095-X. URL https://linkinghub.elsevier.com/retrieve/pii/S037722170300095X.

[5] K. Nurmi and J. Kyngas. Solving scheduling problems for business use using computational intelligence. In *The Fourth International Workshop on Advanced Computational Intelligence*, pages 621–628, Wuhan, China, October 2011. IEEE. ISBN 978-1-61284-375-9 978-1-61284-374-2 978-1-61284-373-5. doi: 10.1109/IWACI.2011.6160083. URL http://ieeexplore.ieee.org/document/6160083/.

[6] Amy Brand, Liz Allen, Micah Altman, Marjorie Hlava, and Jo Scott. Beyond authorship: attribution, contribution, collaboration, and credit. *Learned Publishing*, 28(2):151–155, 2015.

[7] Nysret Musliu, Andrea Schaerf, and Wolfgang Slany. Local search for shift design. *European Journal of Operational Research*, 153(1):51–64, February 2004. ISSN 03772217. doi: 10.1016/S0377-2217(03)00098-5. URL https://linkinghub.elsevier.com/retrieve/pii/S0377221703000985.

[8] Luca Di Gaspero, Johannes Gärtner, Guy Kortsarz, Nysret Musliu, Andrea Schaerf, and Wolfgang Slany. The minimum shift design problem. *Annals of Operations Research*, 155(1):79–105, August 2007. ISSN 0254-5330, 1572-9338. doi: 10.1007/s10479-007-0221-1. URL http://link.springer.com/10.1007/s10479-007-0221-1.

[9] Jorne Van den Bergh, Jeroen Beliën, Philippe De Bruecker, Erik Demeulemeester, and Liesje De Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385, May 2013. ISSN 03772217. doi: 10.1016/j.ejor.2012.11.029. URL https://linkinghub.elsevier.com/retrieve/pii/S0377221712008776.

[10] Lucas Kletzander and Nysret Musliu. Solving the general employee scheduling problem. *Computers & Operations Research*, 113:104794, January 2020. ISSN 03050548. doi: 10.1016/j.cor.2019.104794. URL https://linkinghub.elsevier.com/retrieve/pii/S0305054819302369.

[11] Michael R. Garey and David S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. A Series of books in the mathematical sciences. W. H. Freeman, San Francisco, 1979. ISBN 978-0-7167-1044-8.

[12] James M. Tien and Angelica Kamiyama. On Manpower Scheduling Algorithms. *SIAM Review*, 24(3):275–287, July 1982. ISSN 0036-1445, 1095-7200. doi: 10.1137/1024063. URL `http://epubs.siam.org/doi/10.1137/1024063`.

[13] Hoong Chuin Lau. On the complexity of manpower shift scheduling. *Computers & Operations Research*, 23(1):93–102, January 1996. ISSN 03050548. doi: 10.1016/0305-0548(94)00094-O. URL `https://linkinghub.elsevier.com/retrieve/pii/030505489400094O`.

[14] Dániel Marx. Graph Colouring Problems and Their Applications in Scheduling. *Periodica Polytechnica Electrical Engineering*, 48(1-2):11–16, 2004. doi: N/A. URL `https://pp.bme.hu/ee/article/view/926`.

[15] Peter Brucker, Rong Qu, and Edmund Burke. Personnel scheduling: Models and complexity. *European Journal of Operational Research*, 210(3):467–473, May 2011. ISSN 03772217. doi: 10.1016/j.ejor.2010.11.017. URL `https://linkinghub.elsevier.com/retrieve/pii/S0377221710007897`.

[16] W. Ken Jackson, William S. Havens I, and Harry Dollard. Staff Scheduling: A Simple Approach that Worked. Technical report, 1997.

[17] Tanguy Lapègue, Odile Bellenguez-Morineau, and Damien Prot. A constraint-based approach for the shift design personnel task scheduling problem with equity. *Computers & Operations Research*, 40(10):2450–2465, October 2013. ISSN 03050548. doi: 10.1016/j.cor.2013.04.005. URL `https://linkinghub.elsevier.com/retrieve/pii/S0305054813001056`.

[18] D. Dowling, M. Krishnamoorthy, H. Mackenzie, and D. Sier. Staff rostering at a large international airport. *Annals of Operations Research*, 72:125–147, 1997. ISSN 02545330. doi: 10.1023/A:1018992120116. URL `http://link.springer.com/10.1023/A:1018992120116`.

[19] Tolga Çezik, Oktay Günlük, and Hanan Luss. An integer programming model for the weekly tour scheduling problem: Weekly Tour Scheduling. *Naval Research Logistics (NRL)*, 48(7):607–624, October 2001. ISSN 0894069X. doi: 10.1002/nav.1037. URL `http://doi.wiley.com/10.1002/nav.1037`.

[20] D. Prot, T. Lapègue, and O. Bellenguez-Morineau. A two-phase method for the shift design and personnel task scheduling problem with equity objective. *International Journal of Production Research*, 53(24):7286–7298, December 2015. ISSN 0020-7543, 1366-588X. doi: 10.1080/00207543.2015.1037023. URL `http://www.tandfonline.com/doi/full/10.1080/00207543.2015.1037023`.

[21] Pieter Smet, Andreas T. Ernst, and Greet Vanden Berghe. Heuristic decomposition approaches for an integrated task scheduling and personnel rostering problem. *Computers & Operations Research*, 76:60–72, December 2016. ISSN 03050548. doi: 10.1016/j.cor.2016.05.016. URL `https://linkinghub.elsevier.com/retrieve/pii/S030505481630123X`.

[22] Jonathan F. Bard and Lin Wan. The task assignment problem for unrestricted movement between workstation groups. *Journal of Scheduling*, 9(4):315–341, August 2006. ISSN 1094-6136, 1099-1425. doi: 10.1007/s10951-006-7038-7. URL `http://link.springer.com/10.1007/s10951-006-7038-7`.

[23] Sandjai Bhulai, Ger Koole, and Auke Pot. Simple Methods for Shift Scheduling in Multiskill Call Centers. *Manufacturing & Service Operations Management*, 10(3):411–420, July 2008. ISSN 1523-4614, 1526-5498. doi: 10.1287/msom.1070.0172. URL `http://pubsonline.informs.org/doi/abs/10.1287/msom.1070.0172`.

[24] Monia Rekik, Jean-François Cordeau, and François Soumis. Implicit shift scheduling with multiple breaks and work stretch duration restrictions. *Journal of Scheduling*, 13(1):49–75, February 2010. ISSN 1094-6136, 1099-1425. doi: 10.1007/s10951-009-0114-z. URL `http://link.springer.com/10.1007/s10951-009-0114-z`.

[25] Luca Di Gaspero, Johannes Gärtner, Nysret Musliu, Andrea Schaerf, Werner Schafhauser, and Wolfgang Slany. A Hybrid LS-CP Solver for the Shifts and Breaks Design Problem. In María J. Blesa, Christian Blum, Günther Raidl, Andrea Roli, and Michael Sampels, editors, *Hybrid Metaheuristics*, volume 6373, pages 46–61. Springer Berlin Heidelberg, Berlin, Heidelberg,

2010. ISBN 978-3-642-16053-0 978-3-642-16054-7. doi: 10.1007/978-3-642-16054-7_4. URL http://link.springer.com/10.1007/978-3-642-16054-7_4. Series Title: Lecture Notes in Computer Science.

[26] Quentin Lequy, Guy Desaulniers, and Marius M. Solomon. A two-stage heuristic for multi-activity and task assignment to work shifts. *Computers & Industrial Engineering*, 63(4): 831–841, December 2012. ISSN 03608352. doi: 10.1016/j.cie.2012.05.005. URL https://linkinghub.elsevier.com/retrieve/pii/S0360835212001428.

[27] Marie-Claude Côté, Bernard Gendron, and Louis-Martin Rousseau. Grammar-Based Column Generation for Personalized Multi-Activity Shift Scheduling. *INFORMS Journal on Computing*, 25(3):461–474, August 2013. ISSN 1091-9856, 1526-5528. doi: 10.1287/ijoc.1120.0514. URL http://pubsonline.informs.org/doi/abs/10.1287/ijoc.1120.0514.

[28] Jens O. Brunner and Jonathan F. Bard. Flexible weekly tour scheduling for postal service workers using a branch and price. *Journal of Scheduling*, 16(1):129–149, February 2013. ISSN 1094-6136, 1099-1425. doi: 10.1007/s10951-011-0265-6. URL http://link.springer.com/10.1007/s10951-011-0265-6.

[29] Vincent Boyer, Bernard Gendron, and Louis-Martin Rousseau. A branch-and-price algorithm for the multi-activity multi-task shift scheduling problem. *Journal of Scheduling*, 17(2):185–197, April 2014. ISSN 1094-6136, 1099-1425. doi: 10.1007/s10951-013-0338-9. URL http://link.springer.com/10.1007/s10951-013-0338-9.

[30] Alessandra Parisio and Colin Neil Jones. A two-stage stochastic programming approach to employee scheduling in retail outlets with uncertain demand. *Omega*, 53:97–103, June 2015. ISSN 03050483. doi: 10.1016/j.omega.2015.01.003. URL https://linkinghub.elsevier.com/retrieve/pii/S0305048315000055.

[31] Sana Dahmen and Monia Rekik. Solving multi-activity multi-day shift scheduling problems with a hybrid heuristic. *Journal of Scheduling*, 18(2):207–223, April 2015. ISSN 1094-6136, 1099-1425. doi: 10.1007/s10951-014-0383-z. URL http://link.springer.com/10.1007/s10951-014-0383-z.

[32] Richard Martin Lusby, Troels Martin Range, and Jesper Larsen. A Benders decomposition-based matheuristic for the Cardinality Constrained Shift Design Problem. *European Journal of Operational Research*, 254(2):385–397, October 2016. ISSN 03772217. doi: 10.1016/j.ejor.2016.04.014. URL https://linkinghub.elsevier.com/retrieve/pii/S0377221716302338.

[33] María I. Restrepo, Bernard Gendron, and Louis-Martin Rousseau. Branch-and-Price for Personalized Multiactivity Tour Scheduling. *INFORMS Journal on Computing*, 28(2):334–350, May 2016. ISSN 1091-9856, 1526-5528. doi: 10.1287/ijoc.2015.0683. URL http://pubsonline.informs.org/doi/10.1287/ijoc.2015.0683.

[34] Dori van Hulst, Dick den Hertog, and Wim Nuijten. Robust shift generation in workforce planning. *Computational Management Science*, 14(1):115–134, January 2017. ISSN 1619-697X, 1619-6988. doi: 10.1007/s10287-016-0265-2. URL http://link.springer.com/10.1007/s10287-016-0265-2.

[35] Alex Bonutti, Sara Ceschia, Fabio De Cesco, Nysret Musliu, and Andrea Schaerf. Modeling and solving a real-life multi-skill shift design problem. *Annals of Operations Research*, 252(2): 365–382, May 2017. ISSN 0254-5330, 1572-9338. doi: 10.1007/s10479-016-2175-7. URL http://link.springer.com/10.1007/s10479-016-2175-7.

[36] María I. Restrepo, Bernard Gendron, and Louis-Martin Rousseau. A two-stage stochastic programming approach for multi-activity tour scheduling. *European Journal of Operational Research*, 262(2):620–635, October 2017. ISSN 03772217. doi: 10.1016/j.ejor.2017.04.055. URL https://linkinghub.elsevier.com/retrieve/pii/S0377221717304022.

[37] Sana Dahmen, Monia Rekik, and François Soumis. An implicit model for multi-activity shift scheduling problems. *Journal of Scheduling*, 21(3):285–304, June 2018. ISSN 1094-6136, 1099-1425. doi: 10.1007/s10951-017-0544-y. URL http://link.springer.com/10.1007/s10951-017-0544-y.

[38] Arjan Akkermans, Gerhard Post, and Marc Uetz. Solving the shift and break design problem using integer linear programming. *Annals of Operations Research*, December 2019. ISSN 0254-5330, 1572-9338. doi: 10.1007/s10479-019-03487-6. URL http://link.springer.com/10.1007/s10479-019-03487-6.

[39] Lucas Kletzander and Nysret Musliu. Modelling and Solving the Minimum Shift Design Problem. In Louis-Martin Rousseau and Kostas Stergiou, editors, *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, volume 11494, pages 391–408. Springer International Publishing, Cham, 2019. ISBN 978-3-030-19211-2 978-3-030-19212-9. doi: 10.1007/978-3-030-19212-9_26. URL http://link.springer.com/10.1007/978-3-030-19212-9_26. Series Title: Lecture Notes in Computer Science.

[40] Shuqing Liu, Ting Zhang, Ping Feng, Yali Zheng, and Wenge Chen. Hierarchical Staffing Problem by Shift Design in Nursing Homes: A Two-stage Method. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1013–1018, Hong Kong, Hong Kong, August 2020. IEEE. ISBN 978-1-72816-904-0. doi: 10.1109/CASE48305.2020.9216768. URL https://ieeexplore.ieee.org/document/9216768/.

[41] Pieter Smet, Annelies Lejon, and Greet Vanden Berghe. Demand smoothing in shift design. *Flexible Services and Manufacturing Journal*, March 2020. ISSN 1936-6582, 1936-6590. doi: 10.1007/s10696-020-09380-w. URL http://link.springer.com/10.1007/s10696-020-09380-w.

[42] Sana Dahmen, Monia Rekik, François Soumis, and Guy Desaulniers. A two-stage solution approach for personalized multi-department multi-day shift scheduling. *European Journal of Operational Research*, 280(3):1051–1063, February 2020. ISSN 03772217. doi: 10.1016/j.ejor.2019.07.068. URL https://linkinghub.elsevier.com/retrieve/pii/S0377221719306472.

[43] Vicente Valls, Angeles Pérez, and Sacramento Quintanilla. A graph colouring model for assigning a heterogeneous workforce to a given schedule. *European Journal of Operational Research*, 90 (2):285–302, April 1996. ISSN 03772217. doi: 10.1016/0377-2217(95)00355-X. URL https://linkinghub.elsevier.com/retrieve/pii/037722179500355X.

[44] Mohan Krishnamoorthy and Andreas T. Ernst. The Personnel Task Scheduling Problem. In Panos M. Pardalos, Donald Hearn, Xiaoqi Yang, Kok Lay Teo, and Lou Caccetta, editors, *Optimization Methods and Applications*, volume 52, pages 343–368. Springer US, Boston, MA, 2001. ISBN 978-1-4419-4850-2 978-1-4757-3333-4. doi: 10.1007/978-1-4757-3333-4_20. URL http://link.springer.com/10.1007/978-1-4757-3333-4_20. Series Title: Applied Optimization.

[45] Klaus Jansen. An approximation algorithm for the license and shift class design problem. *European Journal of Operational Research*, 73(1):127–131, February 1994. ISSN 03772217. doi: 10.1016/0377-2217(94)90150-3. URL https://linkinghub.elsevier.com/retrieve/pii/0377221794901503.

[46] Leo G. Kroon, Marc Salomon, and Luk N. Van Wassenhove. Exact and approximation algorithms for the operational fixed interval scheduling problem. *European Journal of Operational Research*, 82(1):190–205, April 1995. ISSN 03772217. doi: 10.1016/0377-2217(93)E0335-U. URL https://linkinghub.elsevier.com/retrieve/pii/0377221793E0335U.

[47] John J Bartholdi III. A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering. *Operations Research*, 29(3):501–510, 1981.

[48] Edmund K Burke, Patrick De Causmaecker, Sanja Petrovic, and Greet Vanden Berghe. Metaheuristics for handling time interval coverage constraints in nurse scheduling. *Applied Artificial Intelligence*, 20(9):743–766, 2006.

[49] Jonathan F Bard and Hadi W Purnomo. Hospital-wide reactive scheduling of nurses with preference considerations. *Iie Transactions*, 37(7):589–608, 2005.

[50] Gareth Beddoe, Sanja Petrovic, and Jingpeng Li. A hybrid metaheuristic case-based reasoning system for nurse rostering. *Journal of Scheduling*, 12(2):99–119, 2009.

[51] Burak Bilgin, Patrick De Causmaecker, Benoît Rossie, and Greet Vanden Berghe. Local search neighbourhoods for dealing with a novel nurse rostering model. *Annals of Operations Research*, 194(1):33–57, 2012.

[52] Edmund K Burke and Tim Curtois. New approaches to nurse rostering benchmark instances. *European Journal of Operational Research*, 237(1):71–81, 2014.

[53] Kimmo Nurmi, Jari Kyngäs, and Nico Kyngäs. The peast algorithm–the key to optimising workforce management and professional sports league schedules. *International Journal of Process Management and Benchmarking 40*, 4(4):406–423, 2014.

[54] Hujin Jin, Gerhard Post, and Egbert van der Veen. Ortec's contribution to the second international nurse rostering competition. In *Proceedings of the 11th international conference on the practice and theory of automated timetabling (PATAT-2016)*, pages 599–501, 2016.

[55] J Gartner, Philip Bohle, Anna Arlinghaus, Werner Schafhauser, Thomas Krennwallner, and Magdalena Widl. Scheduling matters–some potential requirements for future rostering competitions from a practitioner's view. In *12th International Conference of the Practice and Theory of Automated Timetabling*, pages 33–42, 2018.

[56] Jeffrey H Kingston. Khe18: a solver for nurse rostering. In *Proceedings of the of the 12th International Conference on Practice and Theory of Automated Timetabling*, pages 113–127, 2018.

[57] Hylco H Nijp, Debby GJ Beckers, Sabine AE Geurts, Philip Tucker, and Michiel AJ Kompier. Systematic review on the association between employee worktime control and work-non-work balance, health and well-being, and job-related outcomes. *Scandinavian journal of work, environment & health*, pages 299–313, 2012.

[58] The Finnish Institute of Occupational Health. Recommendations for shift work. Technical report, 2019. URL `https://www.ttl.fi/tyontekija/tyoaika/tyoaikojen-kuormittavuuden-arviointi/tyoaikojen-kuormittavuuden-arviointi-jaksotyossa`.

[59] Kati Karhula, Aki Koskinen, Anneli Ojajärvi, Annina Ropponen, Sampsa Puttonen, Mika Kivimäki, and Mikko Härmä. Are changes in objective working hour characteristics associated with changes in work-life conflict among hospital employees working shifts? a 7-year follow-up. *Occupational and environmental medicine*, 75(6):407–411, 2018.

[60] Kati Karhula, Tarja Hakola, Aki Koskinen, Anneli Ojajärvi, Mika Kivimäki, and Mikko Härmä. Permanent night workers sleep and psychosocial factors in hospital work. a comparison to day and shift work. *Chronobiology international*, 35(6):785–794, 2018.

[61] Kimmo Nurmi, Jari Kyngäs, and Nico Kyngäs. Staff rostering optimization: Ideal recommendations vs. real-world computing challenges. In *Intelligent Computing*, pages 274–291. Springer, 2022.

[62] Kimmo Nurmi, Jari Kyngäs, and Nico Kyngäs. Synthesis of employer and employee satisfaction–case nurse rostering in a finnish hospital. *Journal of Advances in Information Technology Vol*, 7 (2), 2016.

[63] Shih-Wei Lin and Kuo-Ching Ying. Minimizing shifts for personnel task scheduling problems: A three-phase algorithm. *European Journal of Operational Research*, 237(1):323–334, August 2014. ISSN 03772217. doi: 10.1016/j.ejor.2014.01.035. URL `https://linkinghub.elsevier.com/retrieve/pii/S0377221714000563`.

[64] Mehran Hojati. A greedy heuristic for shift minimization personnel task scheduling problem. *Computers & Operations Research*, 100:66–76, December 2018. ISSN 03050548. doi: 10.1016/j.cor.2018.07.010. URL `https://linkinghub.elsevier.com/retrieve/pii/S0305054818301941`.

[65] Oğuz Solyali. The Shift Minimization Personnel Task Scheduling Problem: An Effective Lower Bounding Procedure. *Hacettepe Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 34(2), June 2016. ISSN 1301-8752. doi: 10.17065/huniibf.259136. URL `http://dergipark.gov.tr/doi/10.17065/huniibf.259136`.

[66] Davaatseren Baatar, Mohan Krishnamoorthy, and Andreas T. Ernst. A Triplet-Based Exact Method for the Shift Minimisation Personnel Task Scheduling Problem. In Nikhil Bansal

and Irene Finocchi, editors, *Algorithms - ESA 2015*, volume 9294, pages 59–70. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015. ISBN 978-3-662-48349-7 978-3-662-48350-3. doi: 10.1007/978-3-662-48350-3_6. URL http://link.springer.com/10.1007/978-3-662-48350-3_6. Series Title: Lecture Notes in Computer Science.

[67] D. Niraj Ramesh, Mohan Krishnamoorthy, and Andreas T. Ernst. Efficient Models, Formulations and Algorithms for Some Variants of Fixed Interval Scheduling Problems. In Ruhul Sarker, Hussein A. Abbass, Simon Dunstall, Philip Kilby, Richard Davis, and Leon Young, editors, *Data and Decision Sciences in Action*, pages 43–69. Springer International Publishing, Cham, 2018. ISBN 978-3-319-55913-1 978-3-319-55914-8. doi: 10.1007/978-3-319-55914-8_4. URL http://link.springer.com/10.1007/978-3-319-55914-8_4. Series Title: Lecture Notes in Management and Industrial Engineering.

[68] Reshma Chirayil Chandrasekharan, Pieter Smet, and Tony Wauters. An automatic constructive matheuristic for the shift minimization personnel task scheduling problem. *Journal of Heuristics*, February 2020. ISSN 1381-1231, 1572-9397. doi: 10.1007/s10732-020-09439-9. URL http://link.springer.com/10.1007/s10732-020-09439-9.

[69] Tanguy Lapègue. Personnel task scheduling problem library, 2021. URL https://sites.google.com/site/ptsplib/smptsp/instances.

[70] David E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Pub. Co, Reading, Mass, 1989. ISBN 978-0-201-15767-3.

[71] Fred Glover. New Ejection Chain and Alternating Path Methods for Traveling Salesman Problems. In *Computer Science and Operations Research*, pages 491–509. Elsevier, 1992. ISBN 978-0-08-040806-4. doi: 10.1016/B978-0-08-040806-4.50037-X. URL https://linkinghub.elsevier.com/retrieve/pii/B978008040806450037X.

[72] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549, January 1986. ISSN 03050548. doi: 10.1016/0305-0548(86)90048-1. URL https://linkinghub.elsevier.com/retrieve/pii/0305054886900481.

[73] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, May 1983. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.220.4598.671. URL https://www.sciencemag.org/lookup/doi/10.1126/science.220.4598.671.

[74] N. R. M. Kyngäs, K. J. Nurmi, and J. R. Kyngäs. Workforce Scheduling Using the PEAST Algorithm. In Gi-Chul Yang, Sio-Iong Ao, Xu Huang, and Oscar Castillo, editors, *Transactions on Engineering Technologies*, volume 275, pages 359–372. Springer Netherlands, Dordrecht, 2014. ISBN 978-94-007-7683-8 978-94-007-7684-5. doi: 10.1007/978-94-007-7684-5_25. URL http://link.springer.com/10.1007/978-94-007-7684-5_25. Series Title: Lecture Notes in Electrical Engineering.

[75] Kimmo Nurmi, Jari Kyngäs, Dries Goossens, and Nico Kyngäs. Scheduling the Finnish Major Ice Hockey League Using the PEAST Algorithm. In Gi-Chul Yang, Sio-Iong Ao, Xu Huang, and Oscar Castillo, editors, *Transactions on Engineering Technologies*, pages 155–168. Springer Netherlands, Dordrecht, 2015. ISBN 978-94-017-9587-6 978-94-017-9588-3. doi: 10.1007/978-94-017-9588-3_12. URL http://link.springer.com/10.1007/978-94-017-9588-3_12.

[76] Jari Kyngäs, Kimmo Nurmi, Nico Kyngäs, George Lilley, Thea Salter, and Dries Goossens. Scheduling the Australian Football League. *Journal of the Operational Research Society*, 68(8): 973–982, August 2017. ISSN 0160-5682, 1476-9360. doi: 10.1057/s41274-016-0145-8. URL https://www.tandfonline.com/doi/full/10.1057/s41274-016-0145-8.

[77] Jan Christiaens and Greet Vanden Berghe. Slack Induction by String Removals for Vehicle Routing Problems. *Transportation Science*, page trsc.2019.0914, January 2020. ISSN 0041-1655, 1526-5447. doi: 10.1287/trsc.2019.0914. URL http://pubsonline.informs.org/doi/10.1287/trsc.2019.0914.

[78] Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. Timing problems and algorithms: Time decisions for sequences of activities. *Networks*, 65(2):102–128, March 2015. ISSN 00283045. doi: 10.1002/net.21587. URL `http://doi.wiley.com/10.1002/net.21587`.

[79] Øystein Vedaa, Ståle Pallesen, Eilin K Erevik, Erling Svensen, Siri Waage, Bjørn Bjorvatn, Børge Sivertsen, and Anette Harris. Long working hours are inversely related to sick leave in the following 3 months: a 4-year registry study. *International Archives of Occupational and Environmental Health*, 92(4):457–466, 2019.

[80] Kati Karhula, Annina Ropponen, Mikko Härmä, Tarja Hakola, Mia Pylkkönen, Mikael Sallinen, and Sampsa Puttonen. 12 tunnin vuorojärjestelmien turvallinen ja työhyvinvointia edistävä toteuttaminen teollisuudessa. 2016.

[81] Esther M Arkin and Ellen B Silverberg. Scheduling jobs with fixed start and end times. *Discrete Applied Mathematics*, 18(1):1–8, 1987.

Original Publications

Nico Kyngäs, Dries Goossens, Kimmo Nurmi & Jari Kyngäs
Optimizing the unlimited shift generation problem

European Conference on the Applications of Evolutionary Computation
Springer, Berlin, Heidelberg
2012, pp. 508-518

I

Nico Kyngäs, Kimmo Nurmi & Jari Kyngäs
Solving the person-based multitask shift generation problem with breaks

2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)
IEEE
2013, pp. 1-8

II

Kimmo Nurmi, Nico Kyngäs & Jari Kyngäs
Workforce Optimization: the General Task-based Shift Generation Problem

International Journal of Applied Mathematics, 49(4), 2019

III

Nico Kyngäs, Kimmo Nurmi & Dries Goossens
The General Task-based Shift Generation Problem: Formulation and Benchmarks

9th Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2019)
2019, pp. 301-319

IV

Kimmo Nurmi & Nico Kyngäs
A Successful Three-Phase Metaheuristic for the Shift Minimization Personal Task Scheduling Problem

Advances in Operations Research

V

2021

Nico Kyngäs & Kimmo Nurmi
The Extended Shift Minimization Personnel Task Scheduling Problem

Proceedings of the Federated Conference on Computer Science and Information Systems, Annals of Computer Science and Information Systems, Vol. 26, 2021.

VI