Solving the General Employee Scheduling Problem

Lucas Kletzander · **Nysret Musliu**

1 Introduction

In many professions the demand for work requires employees to work in different shifts to cover varying requirements including areas like health care, protection services, transportation, manufacturing or call centers. However, there are many constraints that need to be satisfied in order to create feasible schedules. The demands can be specified in various ways, different legal requirements need to be respected and employee satisfaction has to be taken into account.

Not only is it increasingly difficult to generate schedules by hand for more employees and more requirements, it is also very time consuming. Therefore, automated solutions are mandatory to stay competitive. However, even then it is often hard to provide good solutions in reasonable time as many of the problems are NP-hard.

While not each problem will require the whole set of available restrictions, it is cumbersome to develop a new solver for each problem. Often these can hardly be applied to similar problems differing in some requirements. On the other hand it is a challenging task to provide a general formulation and solution methods that can solve large integrated problems, as even several sub-problems on their own are known to be NP-hard.

The main contribution of this work is the development of a new framework for the General Employee Scheduling problem that allows the implementation of vari-

This work was supported by the Austrian Science Fund (FWF): P24814-N23.
The financial support by the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development is gratefully acknowledged.

L. Kletzander
Christian Doppler Laboratory for Artificial Intelligence and Optimization for Planning and Scheduling, DBAI, TU Wien, Austria
E-mail: lkletzan@dbai.tuwien.ac.at

N. Musliu
Christian Doppler Laboratory for Artificial Intelligence and Optimization for Planning and Scheduling, DBAI, TU Wien, Austria
E-mail: musliu@dbai.tuwien.ac.at

ous heuristic algorithms and their application to a wide range of problems that are specified in the General Employee Scheduling format [4].

In literature general frameworks such as hyper-heuristics [1] exist for solving various problems. However, our framework is specialized for employee scheduling problems and provides new possibilities for the formulation and combination of constraints in this domain.

2 Solver Framework

As the formulation can specify various problems that differ in both the used demands like shift-based, time-based or task-based demand and a wide range of constraints, the focus in the optimization problem will depend on the instance. Therefore, most likely it will be too hard to provide an algorithm that can deal with all problems very well, instead the focus is to provide a possibility to implement different algorithms within the same framework to allow adaptation to various problems as well as increased reusability and reduced additional effort for applying the same algorithm to different problems.

This is possible by providing a unified constraint handling process for easy and independent implementation of new constraints, a common move structure that allows to implement various moves and reuse them in different algorithms and the possibility to design and reuse various algorithms.

The main concept behind the handling of constraints in the framework is to have all constraints obey the same structure of usage by using a common abstract class and a hierarchy of derived classes for specific types of constraints. Then each constraint is treated independently without direct interaction with other constraints, but in a common process that is the same for all constraints. Therefore for each move the relevant constraints can be collected, processed and evaluated in a common way while individual constraints can easily be added, removed or replaced.

Moves are the most important building blocks of any algorithm implemented in this framework. They allow to prepare arbitrary changes to the current solution candidate, to evaluate the impact of those changes by using the constraint mechanisms described before and finally execute or discard the proposed changes depending on the decision from the algorithm.

3 Developing Algorithms, Moves and Constraints

On this basis the framework allows the implementation of algorithms in a general way. The algorithm interface is applied to a problem instance giving access to all the definitions and constraints and a potential solution. This might be an empty schedule or a partial or feasible solution the algorithm is given as a starting point.

An algorithm can use an arbitrary selection of moves. As these moves are independent from the algorithm, they can also be reused in different algorithms. The way algorithms handle their moves and choose which one to evaluate and execute is completely up to the algorithm.

As a proof of concept a new algorithm based on simulated annealing is implemented in the framework with the goal to apply it to several problems from literature as well as to instances from our new instance generator.

The structure of the algorithm is the same for all the problems that are evaluated in this work, only some of the parameters are changed depending on the problem in order to take care of the specific focus of each problem.

A move to be evaluated is selected randomly for each iteration based on different probabilities for each move. The chosen move is then prepared and the effect of the move on the solution value is evaluated. Depending on an acceptance criterion based on the current temperature the move is accepted or aborted.

A range of different moves is implemented to be used with this or other algorithms. These include moves to change shifts, add, remove or exchange shifts, add, remove or exchange task assignments within the shifts as well as scheduling breaks within shifts.

Next to the wide range of constraints specified by our formulation it is also possible to add new constraints like the equity constraint introduced in [5]. In order to apply the simulated annealing algorithm implemented in the framework to this problem, the new constraint could be added with low effort, highlighting the adaptability of the framework.

4 Evaluation

In order to show the applicability to a wide range of problems, we take different problems from literature that cover different types of demand and constraints, translate their instances to our formulation and apply our solver to those instances as well as our own instances.

In particular, these problems include nurse rostering instances [2], the Integrated Task Scheduling and Personnel Rostering Problem (TSPR) [7] with a fixed set of available shift types and task demands as well as the Shift Design Personnel Task Scheduling Problem with Equity Objective (SDPTSP-E) described in [5] and [6] that combines shift design with task demands.

Detailed results are described in the thesis [3]. The results show that these problems coming from different areas of employee scheduling can all be modelled in our formulation and the framework can successfully be applied to all of them. The comparison with the results from literature shows that the implemented general purpose algorithm can provide solid results for most instances across all problems in good runtime and provides a good foundation for the development of more specialized algorithms in the framework.

References

1. Burke, E.K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Qu, R.: Hyper-heuristics: a survey of the state of the art. Journal of the Operational Research Society **64**(12), 1695–1724 (2013). DOI 10.1057/jors.2013.71. URL https://doi.org/10.1057/jors.2013.71
2. Curtois, T.: Employee shift scheduling benchmark data sets. http://www.schedulingbenchmarks.org/ (2017). Accessed: 2018-01-22

3. Kletzander, L.: A heuristic solver framework for the general employee scheduling problem. Master's thesis, TU Wien (2018)
4. Kletzander, L., Mischek, F., Musliu, N., Post, G., Winter, F.: A general modeling format for employee scheduling. Tech. rep., Database and Artificial Intelligence Group, Institut für Informationssysteme, TU Wien (2017). http://www.dbai.tuwien.ac.at/proj/arte/
5. Lapgue, T., Bellenguez-Morineau, O., Prot, D.: A constraint-based approach for the shift design personnel task scheduling problem with equity. Computers & Operations Research **40**(10), 2450–2465 (2013). DOI 10.1016/j.cor.2013.04.005
6. Prot, D., Lapgue, T., Bellenguez-Morineau, O.: 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 (2015). DOI 10.1080/00207543.2015.1037023
7. Smet, P., Ernst, A.T., Vanden Berghe, G.: Heuristic decomposition approaches for an integrated task scheduling and personnel rostering problem. Computers & Operations Research **76**, 60–72 (2016). DOI 10.1016/j.cor.2016.05.016