**Noname manuscript No.**
(will be inserted by the editor)

# From the Field between Theory and Practice in Scheduling

**Gerhard Post**

Received: date / Accepted: date

**Keywords** Timetabling, scheduling

## 1 Introduction

Working as an optimization engineer is an interesting mix between scientific research and practical solutions. Of course when implementing solutions, the first intention is to use the result of scientific research. However, not all scientific research is useful in practice. For example, solving VRPs with 25 jobs by a MILP formulation and a (commercial) solver is not helpful for real life VRPs with 2000 jobs. Moreover, the model needed in practice can be more subtle than scientific research investigates. In his address at the PATAT conference of 2006, Barry McCollum discussed the gap between theory and practice, referring to the field of University Timetabling [1]. At that time, it motivated four attendees to start working on a format to describe High School Timetabling instances, encapsulating many known constraints. The instances from practice were challenging at that time and used in ITC2011 [4]; currently most of these instances can be solved to optimality. It was part of the trend that the competitions became increasingly oriented to practical problems. Thanks to this trend, the research community is studying well-documented instances, closer to practice, and with more practical value.

This contribution is based on my personal experiences of the last 25 years. My work in timetabling and scheduling began in 2001, when I started working for a big OR company. Here I worked in nurse rostering (i.e. shift scheduling), load assignment for routing (loading liquids, or solid 3D loads), multimodal transport, and task assignment. In 2019, I started working at a small company,

Gerhard Post
PCA, Klipperweg 19, 8102 HR Raalte, The Netherlands
and
Department of Applied Mathematics, University of Twente, The Netherlands
E-mail: g.f.post@utwente.nl

mainly addressing service planning, a variant of Multi-Depot VRP. All the time, I kept an appointment at the university, where I researched high school timetabling, sports scheduling and personnel rostering.

My goal with this contribution is to show some challenges one encounters when trying to make an optimization suite serving a wide range of clients. Though some of these challenges look trivial from a scientific point of view, they have to be addressed to be acceptable in practice.

## 2 Areas of attention

### 2.1 Trust and expectations

Before a client begins using an optimization suite, usually several obstacles need to be addressed [2,3]. In large organizations, the person who decides to use the tools could be different from the persons (planners) that will be using it. As a result, there might be some resistance or reluctance from the workplace to start using it. If this hurdle is overcome, the optimization engine may produce solutions that are "good" according to the planner. In my experience this is especially true in personnel rostering, where the assigned shifts directly affect employees' daily lives. This explains why that *self rostering* approaches are frequently used in this domain.

Once the resistance is overcome, there can be planner's rules of thumb that do not always lead to optimality, and hence are violated by your algorithm that is in search for optimality. Moreover "the beam and the mote" effect can occur: seeing the violations of soft constraints in the generated solution, but overlooking these in the manual created one.

### 2.2 Infeasibilities

Usually an empty schedule is feasible (though not always). However, it can happen that the input schedule is not empty, but infeasible. How should this be handled? There are several ways to tackle this issue. One way (as observed in personnel rostering) is to remove shifts, until the schedule is feasible. In other areas this approach might be unfavorable, for example if the time window of the scheduled task is communicated to the customer. In this case the task will remain scheduled with the mark 'tolerated'.

### 2.3 Modeling

No matter how extensive your model is, there will be requests (constraints) for additional ones. These constraints can be trivial in algorithmic sense, but the data to create the constraint could be very customer specific. An example is the constraint that limits the number of tasks in a shift. But then one client requires that depending on the task type, the task is counted in a different way. It is convenient to have a 'customization' mechanism to handle such requests.

### 2.4 Implementation

Optimization processes can run in the background, preferably after midnight. In other situations the available runtime is preferably limited: optimization during working hours wil interfere the work of the planners, and optimization in the evening can interfere with requests that customers[1] can create online. Requests for which the customer is waiting requests are allowed to run for only a few seconds.

For algorithms to be fast, they need specific knowledge on the state of the solution and the details of the change to execute. By contrast, generating a complete solution and passing it to a validator that rechecks all constraints from scratch may be attractive from a coding-simplicity perspective, but it is highly inefficient. Fast implementations make it possible to explore a larger part of the search space, which can significantly improve solution quality.

### 2.5 Support

The optimization call collects data from different (dynamic) sources that lead to the instance to be optimized. It is very helpful to have the full description of the instance saved, say, in a database. When a client asks why the optimization did or didn't do this or that, it is possible to replay the situation. Moreover, bugs (which will hopefully disappear over time) can (hopefully) be reproduced, and the corresponding instance can serve as a test case for the future. Interesting instances can be collected to serve as benchmarks for your algorithms.

## References


1. B. McCollum, *A Perspective on Bridging the Gap Between Theory and Practice in University Timetabling*, PATAT 2006 Brno, Czech Republic, August 30 - September 1, 2006 Revised Selected Papers. Lecture Notes in Computer Science **3867**, Springer, 3 – 23, (2007).
2. K. McKay, W. van Wezel, and T. Waefler, *Expertise in Organizational Planning - Impact on Performance*, The Psychology of Planning in Organizations: Research and Applications, 200 – 228, (2015).
3. S. Petrovic, *"You have to get wet to learn how to swim" applied to bridging the gap between research into personnel scheduling and its implementation in practice*, Annals of Operations Research **275**, 161-179, (2019).
4. Gerhard Post, Luca Di Gaspero, Jeffrey H. Kingston, Barry McCollum, and Andrea Schaerf, *The third international timetabling competition*, Annals of Operations Research **239**, 69–75, (2016).


[1] In service planning it is often the case that a customer can create a visit request.