European Journal of Operational Research 241 (2015) 309–319

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

# Freight railway operator timetabling and engine scheduling

Lukas Bach[a,*], Michel Gendreau[b], Sanne Wøhlk[a]

[a] Cluster for Operations Research and Logistics, Department of Economics and Business, Aarhus University, Aarhus, Denmark
[b] CIRRELT and MAGI, École Polytechnique de Montréal, Montréal, QC, Canada

ARTICLE INFO

*Article history:*
Received 29 July 2013
Accepted 24 August 2014
Available online 22 September 2014

*Keywords:*
Freight railway
Timetabling
Branch-and-Price
Transportation
Scheduling

ABSTRACT

In this paper we consider timetable design at a European freight railway operator. The timetable is designed by choosing the time of service for customer unit train demands among a set of discrete points. These discrete points are all found within the a time-window. The objective of the model is to minimize cost while adhering to constraints regarding infrastructure usage, demand coverage, and engine availability. The model is solved by a column generation scheme where feasible engine schedules are designed in a label setting algorithm with time-dependent cost and service times.



## 1. Introduction

In this paper we consider the problem facing a freight railway operator who has to develop a yearly timetable that includes access to infrastructure. When developing the timetable, a railway operator has to apply for usage of railway infrastructure.

The problem is to schedule a set of unit trains. This is done by fixing the departure/arrival time and assigning an engine to each train. The schedule has to be feasible with respect to infrastructure access. The objective is to minimize driving related costs.

We formulate a model for the problem and solve it using a Branch-and-Price algorithm, which is tested on a set of test instances derived from a real-life case at a freight railway operator.

Deutsche Bahn Schenker Rail Scandinavia (DBSRS) is a railway operator managing transports that originate from and/or have destination in a Scandinavian country. The core business of DBSRS is to provide engines and engine drivers to move customers' cars between stations according to long term contracts. More specifically DBSRS is assigned by its mother companies to handle timetabling and to allocate drivers and engines to the planned trips in the RailNetEurope (RNE) corridor 1 area from Maschen (Hamburg) in the south to Hallsberg (in southern Sweden) in the north.

In Section 2, we describe the problem in detail. In Section 3, we give a literature review. In Section 4.1, we present the modeling of the problem and we describe our solution approach. The case is described in detail in Section 5. Computational experiments are provided in Section 6, and conclusions are drawn in Section 7.

## 2. Problem description

In this section, we describe the problem in general. In Section 5, we describe the case specific details from DBSRS and the network in which they operate.

The overall goal of the problem is to minimize the total costs for the railway operator. When designing a timetable, the total costs include both engine usage, track usage, and other driving related costs. The horizon for the timetable is weekly and has to be repeated for one year. Hence, the demands are serviced at the same time each week during the entire timetable which is valid for one year. The timetables are typically designed more than six months in advance. The contracts between DBSRS and their customers vary in length and scope, but, without loss of generality the contracts all last for at least a year, which covers the timetabling period of a year. Hence, at the time of planning the demand is known.

In Europe, the organization RailNetEurope (RNE) works to harmonize the access to infrastructure in their 38 member countries. When designing their timetables the operators can apply for a train path, which is an origin and destination pair with given departure time and transit times. The paths are designed either by RNE, in which case they are called catalogue paths, or the railway operator can design and apply for paths themselves, in which case they are called tailormade paths. By definition, the catalogue paths comply with the regulations such that the paths are feasible. This paper focuses on the (RNE designed) catalogue paths and thus the design of additional tailormade paths are not considered.

* Corresponding author at: Department of Applied Mathematics, SINTEF ICT, Oslo, Norway. Tel.: +4747963696.

*E-mail addresses:* lukas.bach@sintef.no (L. Bach), michel.gendreau@cirrelt.ca (M. Gendreau), sanw@asb.dk (S. Wøhlk).

http://dx.doi.org/10.1016/j.ejor.2014.08.036
0377-2217/

$$\begin{array}{ll} \min & \textit{Total costs} \\ \text{subject to} & \textit{Demand coverage} \\ & \textit{Availability of engines} \\ & \textit{Replication of weekly timetable} \\ & \textit{Network capacity} \\ & \textit{Path compliance} \\ & \textit{Time-window compliance} \\ & \textit{Flow conservation} \\ & \textit{Engine type compliance} \\ & \textit{Waiting time after demands} \\ & \textit{Transit time between demands} \end{array}$$

**Fig. 1.** Model overview.

An overview of the model can be seen in Fig. 1. In the following we consider the implementation of the constraints and how to handle them.

*Demand coverage*, the demands in this model are for unit trains, i.e., all demands are for a full train using a single engine, that has to drive from an origin to an arrival station. It is not possible to aggregate demand. All demand has to be serviced exactly once.

*Availability of engines*, the model allows for multiple engine types. These differ as regards the demands they can serve, e.g., costs, and safety margins. For each engine type, there is a finite number of engines available.

*Replication of weekly schedule*, the schedule is periodic at a weekly level, and thus it must be possible to replicate it week after week. Therefore we balance the engines in the beginning of the week with the engines at the end of the week. The individual engine does not necessarily start and end at the same station, but the sum of engines at the start and the end of the week must be balanced. How to perform this balance is not straightforward and hence it will be further explained in Section 4.1.

The *network capacity* allows only one train per time-slot. Hence two demands that use the same or have partially overlapping schedules would have to be planned such that no time-slot is used twice.

*Path compliance*, the trains can only be scheduled to start at times where a path exist.

*Time-window compliance*, the start of service of a demand has to be within the time-window and within this time-window we have to select a path which means that it is not enough just to be within the time-window. Therefore any time-window will have an associated set of paths that can be used.

*Flow conservation* ensures that the flows in and out of stations are balanced for all stations of the schedule except at the first and last.

*Engine type compliance* ensures that the correct engine type is used to service a given demand.

*Waiting time after demands* handles the safety margins that are planned after the completion of each demand to avoid propagation of delays. This waiting time is set according to the policy of the railway operator.

*Transit time between demands* is a minimum time that ensure that there is enough time to reposition the engine between servicing two demands. It also includes the necessary time to prepare for service of the latter of these demands. Hence, if it is not possible to reposition an engine between two demands, these cannot be assigned to the same engine.

## 3. Literature

The problem can be formally described as an engine scheduling problem where a set of heterogeneous engines have to serve a set of unit train demands that cannot be serviced simultaneously. Each demand has a time-window during which the service of the demand has to start. Within the time-window service can only start when a time-slot is available. The time-slots are laid out by the infrastructure manager who sets them in advance. There is approximately one time-slot every half hour and with time-windows of about 6 hours, there is a limited number of possible departure times. The planning horizon for the schedule is one week, whereas the rotation period of the engines can be as long as necessary.

General reviews of railway optimization methods can be found in Cordeau, Toth, and Vigo (1998); Huisman, Kroon, Lentink, and Vromans (2005); Lusby, Larsen, Ehrgott, and Ryan (2011). Recent work on railway timetabling can be found in Cacchiani and Toth (2012)

Cacchiani, Caprara, and Toth (2008) consider timetabling in a similar corridor and Ziarati, Soumis, Desrosiers, Gélinas, and Saintonge (1997) solve an engine assignment problem with a heterogeneous fleet, but neither includes the path application concept considered in this paper.

Kuo, Miller-Hooks, and Mahmassani (2010) study the problem of implementing additional paths in a similar timetabling problem with elastic demands.

Nahapetyan and Lawphongpanich (2007) solve a dynamic traffic assignment problem where they use a circular approach as opposed to starting and ending with an empty system. Caimi, Fuchsberger, Laumanns, and Schüpbach (2011) use the Periodic Event Scheduling Model (PESP) on an infrastructure management level to generate feasible time-slot allocations. In the PESP, events are repeated over a time horizon, for instance 1 hour, such that a train is departing at the same minute every hour. Lindner and Zimmermann (2005) use the PESP model to create cost optimal train schedules for a railway operator. As the events in our case are periodic over a week, the time horizon in a PESP model would have to be set to one week. The PESP model considers the design of time-slots or at least the maximum time between different events such as two different trains departing from the same station and the headway between them. In our case this is handled by the choice of train paths and by avoiding overlaps.

Various problems with time-dependent costs have been studied over the past few years. Tagmouti, Gendreau, and Potvin (2007) study a variant of the Capacitated Arc Routing Problem (CARP) where the cost of a route depends on the start time of the service and no waiting is allowed between demands. The problem is transformed to a node routing problem and solved using a column generation scheme. Black, Eglese, and Wøhlk (2013) solve a Price-Collecting Arc Routing Problem (PARP) where transit-time is included in the objective; thus with transit-time being time-dependent the cost becomes so as well. The problem is solved using Variable Neighborhood Search and Tabu Search. Both applications forbid waiting time between demands, whereas our application allows for waiting between demands.

The problem is related to Vehicle Routing Problems with Time Windows (VRPTW). The problem at hand shares the time-window aspects of the VRPTW, but we allow only to service demand at discrete points during the time-window. The access to infrastructure is not considered by the VRPTW. For an introduction to the VRPTW, see Desrosiers, Dumas, Solomon, and Soumis (1995, chap. 2) and for a recent review of the VRPTW that focuses on heuristic methods to solve real-life instances we refer to Bräysy and Gendreau (2005a,b). When not solved by heuristics, the VRPTW is often solved using column generation approaches, see e.g., Kallehauge, Larsen, Madsen, and Solomon (2005). We refer the interested reader to Baldacci, Mingozzi, and Roberti (2011) for a state of the art on solving the VRP and VRPTW. The Discrete Time Window Assignment Vehicle Routing Problem was introduced by Spliet and Desaulniers (2012) and this problem relates to our work as their model chooses among a discrete set of possible time-windows. They solve the problem using a Branch-Price-and-Cut algorithm.

We use a decomposition approach to split the problem into a master problem and a pricing problem. For more information on column generation, we refer the reader to Desrosiers and Lübbecke (2005) and Desrosiers et al. (1995). The pricing problem will be solved as