CASPT 2018 - Full Paper

Rebalancing Rolling Stock by Scheduling Deadhead Trains

Federico Farina · Dennis Huisman · Roberto Roberti · Shadi Sharif Azadeh

May 31, 2018

Abstract This paper deals with the problem of rebalancing the train unit inventory in a railway network. The problem is known in the literature as the *Rolling Stock Rebalancing Problem* (RSRP) and arises in rolling stock rescheduling when the rolling stock plan has been changed due to, for example, planned maintenance and unexpected events. In particular, we investigate the solution of the RSRP by scheduling deadhead trains (i.e., empty trains) in between passenger trains with a hierarchical goal of, first, minimizing the off-balances of train units in the network and, second, minimizing the operational cost for running the deadhead trains. We propose a *Mixed Integer Linear Programming* (MILP) model to formulate the problem and assess its computational performance on real-life instances from the Danish and the Dutch railway networks. The computational results show that the MILP model is able to find the optimal solution of most of the test instances within one hour of computation time and could therefore be a valuable tool to use in practice.

Keywords Railway Planning · Railway Optimization · Rolling Stock Rebalancing · Mixed Integer Linear Programming

Federico Farina
Department of Management Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark
E-mail: fefar@dtu.dk

Dennis Huisman
Econometric Institute and ECOPT, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, the Netherlands
Process Quality and Innovation, Netherlands Railways, the Netherlands
E-mail: huisman@ese.eur.nl, sharifazadeh@ese.eur.nl

Roberto Roberti
Department of Information, Logistics and Innovation, VU Amsterdam, 1081HV Amsterdam, the Netherlands
E-mail: r.roberti@vu.nl

Shadi Sharif Azadeh
Operations Research & Logistics group, Erasmus University Rotterdam, H11-27 Tinbergen Building, Rotterdam, the Netherlands
E-mail: sharifazadeh@ese.eur.nl

1 Introduction

Rolling stock planning is one of the five main phases in railway planning (see Huisman et al. [2005], Caprara et al. [2007], Schöbel [2012]). The other phases are infrastructure planning, line planning, timetable planning, and crew scheduling (see Figure 1). Infrastructure planning deals with long-term decisions, such as extending or upgrading the railway network and builing or expanding stations (see Delorme et al. [2001]). Line planning consists of deciding which lines to operate, at which frequency and the stopping stations of each line. (see Goossens et al. [2004], Schöbel and Scholl [2006], Borndörfer et al. [2007], Schöbel [2012]). Timetable planning defines a timetable based on the line plan defined in the previous step (see Caprara et al. [2002], Peeters [2003], Liebchen and Möhring [2007]). Rolling stock planning provides a *rolling stock (circulation) plan* and is therefore about assigning train units to the lines previously specified accordingly to the timetable previously created (see Schrijver [1993], Fioole et al. [2006]). Finally, crew scheduling decides, for each employee (e.g., train drivers), the tasks that he or she has to complete during the working day (see Caprara et al. [2001], Kroon and Fischetti [2001]). The infrastructure planning step is at a strategic level and involves decisions spanning multiple years. The other four steps are at a tactical or operational level and involve decisions made years or months in advance.

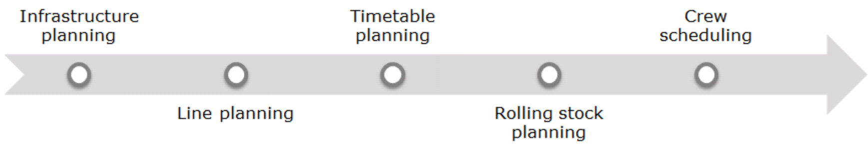

Fig. 1: Five main planning phases in railway planning

The rolling stock plan is constructed once the timetable is defined. Given the expected number of passengers for each train, the rolling stock equipment is assigned to each timetable. Railway companies usually have multiple types of rolling stock equipment available. Most equipment consists of train units. A train unit is composed of a certain amount of coaches that cannot be disconnected when used in the daily operations. When a train is dwelling at a station, units can be coupled or uncoupled to increase or decrease the train capacity of passengers. Once a unit is uncoupled from a train, it is moved to the station's inventory, where operations such as maintenance and cleaning take place. After a certain amount of time, the unit is available and can be coupled with other units. The rolling stock plan involves all operations of coupling and uncoupling train units and storing train units when unused. It is crucial that the train unit inventory of each station is balanced, i.e., the number of units leaving the station (due to coupling operations) are equal to the number of units arriving at the station (due to uncoupling operations). If this condition is violated at a certain station, the station is said to be *off-balanced*.

In passenger railway operations, unexpected events (e.g., infrastructure breakdowns, accidents of various types, rolling stock breakdowns, etc.) or planned maintenance can make part(s) of the railway network unavailable. This could make the timetable, the rolling stock plan, or the crew plan unrealizable as previously planned. During these unforeseen events, the daily plans have to be rescheduled in order to cope with the new network and equipment conditions (for an exhaustive description of this process, the reader is referred to Jespersen-Groth et al. [2009]).

The rolling stock rescheduling problem aims at reconstructing a feasible rolling stock plan after a disruption - for a general overview of recovery models for real-time railway rescheduling see Cacchiani et al. [2014]. Within this context, the *Rolling Stock Rebalancing Problem* (RSRP) is a fundamental component of rolling stock rescheduling that aims at resolving off-balances of train units at station (see Budai et al. [2010]); these off-balances arise whenever there are still off-balanced stations in the network once the rescheduling of the rolling stock has taken place. This paper focuses on the problem of resolving off-balances, which can occur both in short-term planning and in real-time operations, by scheduling *deadhead trains* (i.e., empty trains scheduled within passenger trains). Hereafter, we call this planning problem *Rolling Stock Rebalancing Problem with Deadhead Trains* (RSRP-DT). The main contributions of the paper are twofold: (*i*) we introduce a *Mixed Integer Linear Programming* (MILP) model to formulate and solve the RSRP-DT, and (*ii*) we provide computational evidence of the effectiveness of the proposed MILP model by solving realistic instances based on the Danish and the Dutch railway networks.

The paper is organized as follows. In Section 2, we introduce the RSRP-DT in detail. In Section 3, the relevant literature is reviewed. In Section 4, we describe the MILP proposed to solve the RSRP-DT. In Section 5, we describe the Danish and Dutch railway networks used to test the model. In Section 6, we present and discuss the computational results achieved by the MILP model on the instances. In Section 7, we discuss some conclusions and outline future research perspectives.

2 Problem Description

In this section, we introduce the RSRP-DT in detail. In particular, we introduce the problem input, outline how deadhead trains can resolve off-balances, define the objectives of the RSRP-DT, and provide a small railway network to better illustrate the RSRP-DT.

Problem input. As previously mentioned, in order to carry out a rolling stock plan it is crucial that the rolling stock balancing condition is not violated over the entire railway network. Nevertheless, due to events that can interfere with the daily operations (e.g., planned maintenance, disruptions, etc), there can be off-balanced stations that have a surplus or a deficit in the train unit inventory. We say that a station's inventory has a surplus of a train units (of a specific rolling stock type) if, at a certain point in time, the number of train units exceeds the planned number of train units of that type at that station (at that particular point in time). Similarly, we say that a station's inventory has a deficit if, at a certain point in time, the number of train units is lower than the number of planned train units of that type at that station. It is clear that, over the entire railway network, for each rolling stock type the total surplus of train units equals the total deficit of train units of that type. Because each surplus and each deficit is related to a point in time, part of the input of the RSRP-DT is given by the list of station's inventories with a surplus and with a deficit, the amount of off-balanced units at each off-balanced station, and a time representing when that surplus or deficit begins. Moreover, as surpluses and deficits must be solved within a certain time period, an additional time indicating when each surplus and each deficit has to be resolved is also provided. For each off-balanced, it is therefore possible to define a time window within which the off-balance begins and should also be resolved.

The timetable of the passenger trains are also part of the input. They are assumed to be fixed. Indeed, in the RSRP-DT we consider, it is allowed to schedule deadhead trains in between passenger trains without interfering with them (i.e., no changes are allowed to the passenger trains' timetables). Since deadhead trains will be scheduled in between passengers trains, some of the classic timetabling constraints are considered in the model. First, the minimum headway time between trains has to be guaranteed - the headway time is a measurement of the distance or time between trains in a railway system (in practice, it varies between 3 and 5 minutes). Second, station and track capacities are considered; there has to be enough capacity (i.e., available free platforms) to accommodate deadhead trains at each station along their routes. In order to model capacity constraints, information about the physical railway network considered, in particular the stations capacities and the number of tracks connecting stations, are also part of the input.

Deadhead trains. To rebalance the rolling stock surpluses and deficits in the network, new deadhead trains will be scheduled. Each deadhead train starts from a station that has a surplus of rolling stock units of a certain type and ends at a station where there is a deficit of that type of units. The route of each deadhead is not fixed a-priori. The travel time between stations is given and fixed. The deadhead trains can pass by the stations along their path and dwell there (if needed) without exceeding a given maximum dwell time at each station. The freedom in scheduling deadhead trains is limited by the scheduled passenger timetable, which dictates when it is allowed to pass by or dwell at each station of the network. Each deadhead train consists of a certain number of train units, which depends on the surplus and the deficit of the starting and ending station, respectively. In particular, a deadhead train can never consist of more train units than the surplus of the starting station and the deficit of the ending station. We assume that the number of train unit of each deadhead train does not change along its route, so it can solve at most one surplus and one deficit.

Objective. The objective of the RSRP-DT is twofold. First, the number of off-balances has to be minimized (notice that it might not be possible to solve all of them due to time windows, headway, and capacity constraints), and second the total travel time (including the dwell time) of the scheduled deadhead trains has to be minimized. The first objective has clearly higher priority since off-balances prevent the realization of the rolling stock plan. The second objective is due to the fact that the operational costs of the new scheduled trains are directly proportional to their travel times; moreover, since deadhead trains are scheduled in between passengers trains, it is preferable to have short routes for deadhead trains in order to avoid potential extra delays.

A real-life example. Figure 2 (taken from Bešinović et al. [2016]) shows a portion of the Dutch railway network containing five main stations (Eindhoven, Nijmegen, Utrecht, 's-Hertogenbosch and Tilburg) and twenty smaller stations. Let's assume that we have four off-balances for a given train unit type: two surpluses of one unit in Utrecht and two units in Nijmegen, and two deficits of two units in Tilburg and one unit in Eindhoven. For each off-balanced station, the railway operator defines a time window to resolve the off-balance; let's assume the time windows are 20:00-22:00 (in Utrecht and Nijmegen) and 20:30-23:00 (in Eindhoven and Tilburg).

The off-balances can be solved by scheduling two deadhead trains: the first that carries one train unit, starts from Utrecht at 20:05, and arrives in Eindhoven at 20:55, and the second that carries two train units, starts from Nijmegen at 21:30, and arrives in Tilburg at 22:32.

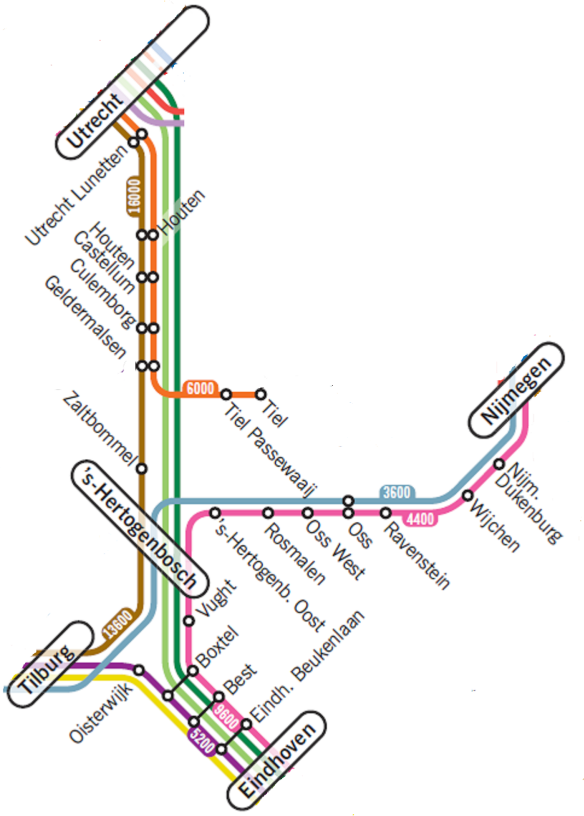

Fig. 2: Dutch subnetwork example

3 Literature Review

To the best of our knowledge, the first paper dealing with the RSRP is owed to Budai et al. [2010], where the RSRP is addressed as a short-term planning problem. The main inputs of the considered RSRP are (*i*) the train timetables of a given planning horizon, (*ii*) the available rolling stock, (*iii*) the current rolling stock plan, (*iv*) the rolling stock off-balances, and (*v*) the underlying railway network. The goal of the considered RSRP is to modify the input rolling stock plan in order to minimize the number of rolling stock off-balances at the end of the planning horizon. The authors describe two heuristic algorithm to solve this problem. Computational results achieved on real-life instances provided by NS (the main Dutch railway operator) show that the proposed heuristics can resolve most of the off-balances in short amount of computation times.

The contributions of Nielsen et al. [2012], Cacchiani et al. [2012], Haahr et al. [2016], Lusby et al. [2017] address the RSRP in the wider context of rolling stock rescheduling.

Nielsen et al. [2012] present a rolling horizon solution approach for real-time rolling stock rescheduling after a disruption. They propose to periodically reschedule, over a limited rolling horizon, the rolling stock plan in order to take into account new disruptions. In particular, a MILP model similar to the model presented in Fioole et al. [2006] is used to reschedule the rolling stock over the planning horizon by considering the train timetables adjusted after the disruptions. The model aims at minimizing the differences between the new and the original rolling stock plans in terms of number of canceled trips, rolling stock off-balances at the end of the planning horizon, and additional required shunting operations. The proposed model is tested on real-life instances considering a few lines of the Dutch railway network and a planning horizon of up to three hours. Computational results show that the proposed model can solve these instances in a few seconds of computation time.

Cacchiani et al. [2012] extend the model of Fioole et al. [2006] by presenting a two-stage optimization model to construct a robust rolling stock plan for passenger trains. In the first stage, a robust rolling stock plan is created using a heuristic algorithm. In the second stage, a set of disruption scenarios are used to create optimal recovery action responses and analyze them. The computational results, based on instances from the Dutch railway operator NS, show that the generated robust rolling stock plans have slightly higher nominal costs but are easier to recover (and have lower operational costs) than non-robust rolling stock plans.

Lusby et al. [2017] develop a unit-based model for rescheduling rolling stock of passenger trains, where the paths of each rolling stock unit are generated separately. Each path can include specific constraints related to the rolling stock unit such as maintenance restrictions. The model is solved by applying column generation and tested on real-life instances from the Danish suburban train network. The computation results indicate that near-optimal solutions can be provided quickly with the proposed column-generation-based method.

Haahr et al. [2016] compare the computational results of two rolling stock rescheduling models. The first model is based on the model of Lusby et al. [2017] extended to consider the order of the train units, and it is solved exactly through a column-and-row generation method. The second model has been presented in Nielsen et al. [2012] and solved with CPlex. The computational results are based on instances from the Dutch railway network and the Danish suburban network. The authors show that both methods are fast enough to be used in real-time optimization, but the column-and-row generation method has higher computational times. However, the authors argue that the column-and-row generation method can more easily accommodate unit specific constraints (e.g., maintenance related constraints).

The recent contributions of Cadarso et al. [2013], Kroon et al. [2015], Veelenturf et al. [2017], Wagenaar et al. [2017] address the rolling stock rescheduling problem by focusing on passengers.

Cadarso et al. [2013] present an optimization problem that integrates timetabling and rolling stock rescheduling and that considers dynamic passenger demands. The solution approach is an iterative process. In the first step of the process, the potential passenger demand is computed; this passenger demand is based on the frequency of the lines only as the timetable are adjusted in the second phase. In the second step, both the timetabling and the rolling stock rescheduling problem are formulated as a single integrated MILP model; this model allows to cancel existing trains, schedule new trains, but does not allow to reschedule existing trains. Computational results on the rapid transit network of Madrid are presented. The computational times taken to solve the proposed model vary from a few seconds to a few minutes, which makes the approach usable to solve real-life scenarios.

Kroon et al. [2015] propose an iterative heuristic algorithm to solve the real-time rolling stock rescheduling problem with dynamic passenger flows. The goal of the algorithm is to improve the service offered to passengers by minimizing the number of canceled trains and the total passenger delays. At each iteration, the rolling stock is rescheduled by using a MILP model, and the passenger flows are simulated. Computational results show that real-life instances from the Dutch railway network can be solved in a few minutes of computation time.

Veelenturf et al. [2017] address an integrated rolling stock rescheduling and timetabling problem considering the changed passenger demand with the goal of improving the overall service for the passengers. The timetabling problem is limited to schedule extra stops (i.e., stops at stations where trains are not scheduled to dwell) in the planned train timetables. An iterative heuristic approached is described. The main idea of the heuristic is to iteratively generate new train schedules and assess the quality from the passenger's perspective of the resulting timetable by running a simulation model that decides the paths of the passengers. They present results based on real-life instances from the Dutch railway network showing that the service can be improved for the passenger without increasing the rolling stock rescheduling costs.

Wagenaar et al. [2017] address the RSRP by considering deadhead trains and adjusted passengers demand. They propose a MILP formulation of the problem that allows first to reschedule the rolling stock considering the passenger demand after the disruption and second to schedule a fixed set of deadhead trips given as input to the model. The set of potential deadhead trains is constructed with a method that makes several assumptions to limit the size of the potential deadhead trains. The model is solved by using CPlex. Computational results on different versions of the problem (with and without adjusted passenger demand and with and without deadhead trips) show that instances from the Dutch network that be solved in a few minutes of computation time.

4 A Mathematical Model for the RSRP-DT

In this section, we formulate the MILP model for the RSRP-DT. First, we formally introduce the RSRP-DT. Then, we describe the decision variables of the model. Finally, we provide the model and describe it in detail.

Formal Description of the RSRP-DT. The underlying physical railway network can be represented as a directed graph $G = (V, A)$. The vertex set V represents the set of all stations of the network. The arc set A represents the tracks that connect adjacent stations, i.e., each arc $(i, j) \in A$ represents the track from station $i \in V$ to station $j \in V$.

A maximum dwell time $\overline{\delta}_i$ is associated with each station $i \in V$, and a fixed travel time tt_{ij} is associated with each arc $(i, j) \in A$. A minimum headway time σ has to pass between two consecutive trains travel along the same arc.

The set of stations having a surplus of rolling stock units is indicated as $V^+ \subset V$, and the surplus of rolling stock units of station $i \in V^+$ is represented by coefficient a_i. Similarly, the set of stations having a deficit of rolling stock units is indicated as $V^- \subset V$, and the deficit of rolling stock units of station $i \in V^-$ is represented by coefficient b_i. Let $P = \{(i, j) \,|\, i \in V^+, j \in V^-\}$ be the set of all pairs of stations (i, j), where i is a surplus station and j is a deficit station.

A time window $[tws_i, twe_i]$ is also associated with each surplus and each deficit station $i \in V^+ \cup V^-$ and represents the earliest and the latest possible times within which the off-balance should be resolved.

Part of the input of the problem is given by the timetable of the passenger trains. As the passenger timetable cannot be changed, it defines the instants of time when a deadhead train can pass through a station and travel along an arc. In particular, for each station $i \in V$, a set K_i of time intervals where a deadhead train can pass through and dwell at station i is defined as follows $K_i = \{[e_i^1, l_i^1], [e_i^2, l_i^2], \ldots, [e_i^{|K_i|}, l_i^{|K_i|}]\}$, where $e_i^k \le l_i^k$ ($k = 1, \ldots, |K_i|$) and $l_i^k + 1 < e_i^{k+1}$ ($k = 1, \ldots, |K_i| - 1$). Moreover, for each arc $(i, j) \in A$, a set Q_{ij} of time intervals where a deadhead train can travel along the arc is defined as $Q_{ij} = \{[e_{ij}^1, l_{ij}^1], [e_{ij}^2, l_{ij}^2], \ldots, [e_{ij}^{|Q_{ij}|}, l_{ij}^{|Q_{ij}|}]\}$, where $e_{ij}^q \le l_{ij}^q$ ($q = 1, \ldots, |Q_{ij}|$), and $l_{ij}^q + 1 < e_{ij}^{q+1}$ ($q = 1, \ldots, |Q_{ij}| - 1$).

The goal of the RSRP-DT is to schedule deadhead trains in order to minimize, first, the number of off-balances (i.e., surplus at stations of the set V^+ and deficits at stations of the set V^-) at the end of the planning horizon and, second, the travel time of the deadhead trains. The scheduled deadhead trains have to be scheduled in such a way that (i) the minimum headway time between consecutive trains (both passenger and deadhead train) leaving a station or traveling along an arc is satisfied, (ii) there is enough capacity at each station to accommodate all passenger and deadhead trains, (iii) trains do not overtake each other, and (iv) off-balances are resolved within the given time windows.

Decision Variables. The mathematical model proposed in this paper uses the following decision variables:

$x_{k\ell}^{ij} \in \{0, 1\}$	binary variable equal to 1 if the path from surplus station $i \in V^+$ to deficit station $j \in V^-$ traverses arc $(k, \ell) \in A$ (0 otherwise);
$u_k^{ij} \in \mathbb{R}_+$	arrival time at station $k \in V$ of deadhead train sent from station $i \in V^+$ to station $j \in V^-$;
$v_k^{ij} \in \mathbb{R}_+$	departure time from station $k \in V$ of deadhead train sent from station $i \in V^+$ to station $j \in V^-$;
$\delta_k^{ij} \in \mathbb{R}_+$	auxiliary variable indicating the dwell time at station $k \in V \setminus \{i, j\}$ of deadhead train sent from station $i \in V^+$ to station $j \in V^-$;
$z^{ij} \in \mathbb{Z}_+$	number of rolling stock units sent from station $i \in V^+$ to station $j \in V^-$;
$\xi_i^+ \in \mathbb{Z}_+$	non-negative integer variable representing the surplus of station $i \in V^+$ after rebalancing;
$\xi_i^- \in \mathbb{Z}_+$	non-negative integer variable representing the deficit of station $i \in V^-$ after rebalancing;
$g_k^{ijrs} \in \{0, 1\}$	binary variable equal to 1 if the deadhead train sent from station $r \in V^+$ to station $s \in V^-$ arrives at station $k \in V$ after the deadhead train sent from station $i \in V^+$ to station $j \in V^+$ has left station k, if both trains go through station k (0 otherwise);
$h_{k\ell}^{ijrs} \in \{0, 1\}$	binary variable equal to 1 if the deadhead train sent from station $r \in V^+$ to station $s \in V^-$ travels along arc $(k, \ell) \in A$ after the deadhead train sent from station $i \in V^+$ to station $j \in V^-$ travels along the same arc (k, ℓ), if both trains go through arc (k, ℓ) (0 otherwise);
$y_{k\alpha}^{ij} \in \{0, 1\}$	binary variable equal to 1 if the deadhead train from station $i \in V^+$ to station $j \in V^-$ passes through (and possibly dwells at) station $k \in V \setminus \{i, j\}$ in the time interval $[e_k^\alpha, l_k^\alpha]$, where $\alpha = 1, \ldots, |K_k|$ (0 otherwise);
$w_{k\ell\alpha}^{ij} \in \{0, 1\}$	binary variable equal to 1 if the deadhead train from station $i \in V^+$ to station $j \in V^-$ travels arc $(k, \ell) \in A$ within the time interval $[e_{ij}^\alpha, l_{ij}^\alpha]$, where $\alpha = 1, \ldots, |Q_{ij}|$ (0 otherwise).

Mathematical Model. Now, the RSRP-DT can be formulated as follows:

$$\min \quad M\left(\sum_{i \in V^+} \xi_i^+ + \sum_{i \in V^-} \xi_i^- \right) +$$

$$+\sum_{i\in V^+}\sum_{j\in V^-}\left(\sum_{(k,\ell)\in A} tt_{k\ell}x_{k\ell}^{ij}+\sum_{k\in V\setminus\{i,j\}}\delta_k^{ij}\right) \tag{1}$$

$$\text{s.t.} \sum_{j\in V^-} z^{ij}+\xi_i^+ = a_i \qquad i\in V^+ \tag{2}$$

$$\sum_{i\in V^+} z^{ij}+\xi_j^- = b_j \qquad j\in V^- \tag{3}$$

$$z^{ij}\le \min\{a_i,b_j\}\sum_{(i,k)\in A} x_{ik}^{ij} \qquad (i,j)\in P \tag{4}$$

$$\sum_{(k,\ell)\in A} x_{k\ell}^{ij}\le 1 \qquad (i,j)\in P \quad k\in V\setminus\{j\} \tag{5}$$

$$\sum_{(k,\ell)\in A} x_{k\ell}^{ij} = \sum_{(\ell,k)\in A} x_{\ell k}^{ij} \qquad (i,j)\in P \quad k\in V\setminus\{i,j\} \tag{6}$$

$$\sum_{(k,i)\in A} x_{ki}^{ij}=0 \qquad (i,j)\in P \tag{7}$$

$$\sum_{(j,\ell)\in A} x_{j\ell}^{ij}=0 \qquad (i,j)\in P \tag{8}$$

$$v_k^{ij}+tt_{k\ell}\le u_\ell^{ij}+M(1-x_{k\ell}^{ij}) \qquad (i,j)\in P \quad (k,\ell)\in A \tag{9}$$

$$v_k^{ij}+tt_{k\ell}\ge u_\ell^{ij}-M(1-x_{k\ell}^{ij}) \qquad (i,j)\in P \quad (k,\ell)\in A \tag{10}$$

$$v_k^{ij}-u_k^{ij}=\delta_k^{ij} \qquad (i,j)\in P \quad k\in V\setminus\{i,j\} \tag{11}$$

$$u_k^{ij}\ge v_k^{rs}+1-M\Big(2-\sum_{(k,\ell)\in A}(x_{k\ell}^{ij}+x_{k\ell}^{rs})+g_k^{ijrs}\Big) \qquad (i,j),(r,s)\in P \quad k\in V \tag{12}$$

$$u_k^{rs}\ge v_k^{ij}+1-M\Big(3-\sum_{(k,\ell)\in A}(x_{k\ell}^{ij}+x_{k\ell}^{rs})-g_k^{ijrs}\Big) \qquad (i,j),(r,s)\in P \quad k\in V \tag{13}$$

$$v_k^{ij}\ge v_k^{rs}+\sigma-M(2-x_{k\ell}^{ij}-x_{k\ell}^{rs}+h_{k\ell}^{ijrs}) \qquad (i,j),(r,s)\in P \quad (k,\ell)\in A \tag{14}$$

$$v_k^{rs}\ge v_k^{ij}+\sigma-M(3-x_{k\ell}^{ij}-x_{k\ell}^{rs}-h_{k\ell}^{ijrs}) \qquad (i,j),(r,s)\in P \quad (k,\ell)\in A \tag{15}$$

$$\sum_{(k,j)\in A} tws_j x_{kj}^{ij}\le u_j^{ij}\le \sum_{(k,j)\in A} twe_j x_{kj}^{ij} \qquad (i,j)\in P \tag{16}$$

$$\sum_{(i,\ell)\in A} tws_i x_{i\ell}^{ij}\le v_i^{ij}\le \sum_{(i,\ell)\in A} twe_i x_{i\ell}^{ij} \qquad (i,j)\in P \tag{17}$$

$$\sum_{\alpha=1}^{|K_k|} e_k^\alpha y_{k\alpha}^{ij}\le u_k^{ij} \qquad (i,j)\in P \quad k\in V\setminus\{i,j\} \tag{18}$$

$$v_k^{ij}\le \sum_{\alpha=1}^{|K_k|} l_k^\alpha y_{k\alpha}^{ij} \qquad (i,j)\in P \quad k\in V\setminus\{i,j\} \tag{19}$$

$$\sum_{\alpha=1}^{|K_k|} y_{k\alpha}^{ij}\le \sum_{(k,\ell)\in A} x_{k\ell}^{ij} \qquad (i,j)\in P \quad k\in V\setminus\{i,j\} \tag{20}$$

$$\sum_{(\ell,k)\in A}\sum_{\alpha=1}^{|Q_{\ell k}|} l_{\ell k}^\alpha w_{\ell k\alpha}^{ij}\ge u_k^{ij} \qquad (i,j)\in P \quad k\in V \tag{21}$$

$$v_k^{ij} \geq \sum_{(k,\ell)\in A} \sum_{\alpha=1}^{|Q_{k\ell}|} e_{k\ell}^{\alpha} w_{k\ell\alpha}^{ij} \qquad (i,j) \in P \quad k \in V \tag{22}$$

$$\sum_{\alpha=1}^{|Q_{k\ell}|} w_{k\ell\alpha}^{ij} \leq x_{k\ell}^{ij} \qquad (i,j) \in P \quad (k,\ell) \in A \tag{23}$$

$$\delta_k^{ij} \leq \overline{\delta}_k \qquad (i,j) \in P \quad k \in V \setminus \{i,j\} \tag{24}$$

$$\xi_i^+ \in \mathbb{Z}_+ \qquad i \in V^+ \tag{25}$$

$$\xi_i^- \in \mathbb{Z}_+ \qquad i \in V^- \tag{26}$$

$$x_{k\ell}^{ij} \in \{0,1\} \qquad (i,j) \in P \quad (k,\ell) \in A \tag{27}$$

$$u_k^{ij}, v_k^{ij} \in \mathbb{R}_+ \qquad (i,j) \in P \quad k \in V \tag{28}$$

$$z^{ij} \in \mathbb{Z}_+ \qquad (i,j) \in P \tag{29}$$

$$g_k^{ijrs} \in \{0,1\} \qquad (i,j),(r,s) \in P \quad k \in V \tag{30}$$

$$h_{k\ell}^{ijrs} \in \{0,1\} \qquad (i,j),(r,s) \in P \quad (k,\ell) \in A \tag{31}$$

$$y_{k\alpha}^{ij} \in \{0,1\} \qquad (i,j) \in P \quad k \in V \setminus \{i,j\} \quad \alpha = 1,\ldots,|K_k| \tag{32}$$

$$w_{k\ell\alpha}^{ij} \in \{0,1\} \qquad (i,j) \in P \quad (k,\ell) \in A \quad \alpha = 1,\ldots,|Q_{ij}| \tag{33}$$

The objective function (1) aims at minimizing, first, the off-balance of the surplus and deficit stations once the deadhead trains have been scheduled, and, second, the total travel and dwell time of the scheduled deadhead trains.

Constraints (2) and (3) set the variables ξ_i^+ and ξ_i^- equal to the final off-balance of the surplus and deficit stations, respectively. Constraints (4) allow to set the number of train units to form a deadhead train from station $i \in V^+$ to station $j \in V^-$.

Constraints (5) ensure that each station is visited at most once by the deadhead train from station $i \in V^+$ to station $j \in V^-$. Constraints (6) are flow conservation constraints for each station and each pair of surplus/deficit $(i,j) \in P$. Constraints (7) ensure that any deadhead train originating from station $i \in V^+$ does not return to station i. Similarly, constraints (8) ensure that any deadhead train terminating at station $j \in V^-$ does not leave from station j.

Constraints (9) and (10) guarantee that, if the deadhead train from station $i \in V^+$ to station $j \in V^-$ travels along arc $(k,\ell) \in A$, then the arrival time at station ℓ is equal to the departure time from station k plus the travel time from k to ℓ, that is, $u_\ell^{ij} = v_k^{ij} + tt_{k\ell}$. Constraints (11) set variables δ_k^{ij} equal to the dwell time at station $k \in V \setminus \{i,j\}$ of the deadhead train from $i \in V^+$ to $j \in V^-$.

Constraints (12)-(15) guarantee that the scheduled deadhead trains do not violate headway time constraints among each other. This is guaranteed by making sure that the visits at a station of two deadhead trains do not overlap, and neither does the traversing of an arc of two deadhead trains.

Constraints (16)-(17) ensure that the departure from station $i \in V^+$ and the arrival at the station $j \in V^-$ of the deadhead train from i to j are scheduled within the corresponding time windows. Constraints (18)-(20) model that the arrival and the departure time at a station $k \in V \setminus \{i,j\}$ of the deadhead train from $i \in V^+$ to $j \in V^-$ lie within one of the feasible time intervals of the set K_k; this ensures that the capacity of each station is satisfied. Similarly, constraints (21)-(23) model that the time at which a deadhead train traverses each arc of the network is compatible with the feasible time intervals $Q_{k\ell}$.

Finally, constraints (24)-(33) define the range of the decision variables.

5 Test Instances

In order to test the computational performance of the model presented in Section 4, we generated a set of 160 benchmark instances based on the Danish and the Dutch railway networks. We have collected full information on the railway networks of both countries. The main railway operations of the two countries (i.e., DSB for Denmark, and NS for the Netherlands) provided us with the actual passenger's timetables of a standard week day. Figure 3 and 4 show the Danish and Dutch network, respectively.

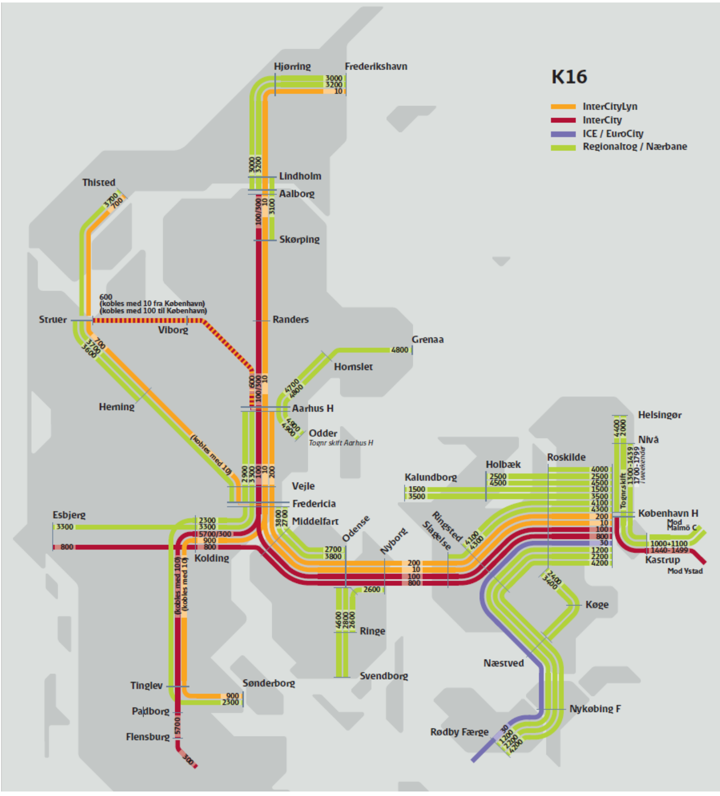

Fig. 3: 2016 Danish train lines

To generate a comprehensive set of benchmark instances, we have identified a set of parameters and different values for each of them. First, we define a set of stations $O \subseteq V$ that can be off-balanced. Then, the first two parameters we define are the number of surplus stations ($|V^+|$) and the number of deficit stations ($|V^-|$) in the network, where $V^+, V^- \subset O$. Once the cardinality of the sets V^+ and V^- are decided, the actual stations of the two sets are randomly selected from the set O (the random function has a uniform distribution). Then, the minimum $\underline{o}$ and the maximum $\overline{o}$ off-balance of any station $i \in V^+ \cup V^-$ is defined, and the off-balances are randomly generated in such a way that $\underline{o} \leq a_i \leq \overline{o}$ $(i \in V^+)$, $\underline{o} \leq b_i \leq \overline{o}$ $(i \in V^-)$, and $\sum_{i \in V^+} a_i = \sum_{i \in V^-} b_i$. To generate a time window for each surplus and each deficit station, we use two parameters $\underline{tws}$ and $\overline{twe}$ that represent a generic time window

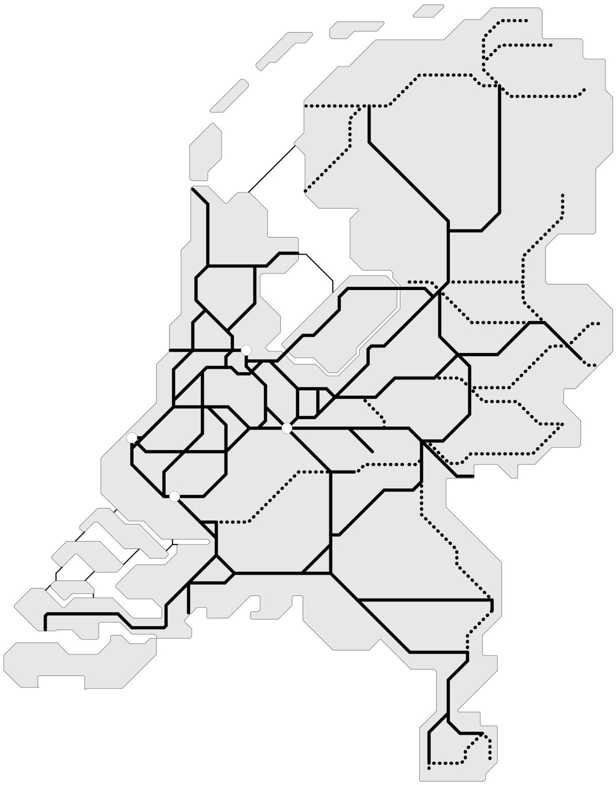

Fig. 4: 2017 Dutch train lines

(in minutes); for each off-balanced station, the actual time window is generated by adding a random (with uniform distribution) value between -60 and +60 to $\underline{tws}$ and $\overline{twe}$. For example if $\underline{tws}$ is set to 18:00 and $\overline{twe}$ is set to 02:00, the resulting time window for a station could be 17:30 ($\underline{tws}$ − 30 minutes) and 02:15 ($\overline{twe}$ + 15 minutes). The last parameter is the maximum dwell time $\overline{\delta}$ allowed for all deadhead trains at any station $i \in V$ - we assume $\overline{\delta} = \overline{\delta}_i$ for each $i \in V$. Therefore, to summarize, the set of parameters used to generate the instances are the following:

– Number of surplus stations $|V^+|$ and number of deficit stations $|V^-|$
– Minimum $\underline{o}$ and maximum $\overline{o}$ amount of each surplus and each deficit
– Time window start $\underline{tws}$ and time window end $\overline{twe}$
– Maximum dwell time $\overline{\delta}_i$ $(i \in V)$

We generate instances on both the Danish and Dutch network using combinations of the following parameter values: $|V^+| \in \{3, 5\}$, $|V^-| \in \{3, 5\}$, $\underline{o} = 1$, $\overline{o} = 3$, $\underline{tws} \in \{13{:}00, 18{:}00\}$, $\overline{twe} \in \{2{:}00\}$, and $\overline{\delta} \in \{10, 25\}$. These values assigned to the parameters are based on discussions with both the Dutch and Danish railway operators and are guided by historical data. It should be noted that it is assumed that within each surplus station time window there is a driver available to drive a deadhead train (according to the railway operators this is a realistic assumption).

For some parameters we consider multiple values to allow us to create instances with different features and to evaluate the performance of our formulation more thoroughly. There are 16 possible combinations of the parameters. Since in the generation of the instances there is a random component (e.g., in the choice of the off-balances and in the time windows), we have generated five different instances using the same configuration of parameters for each of the two networks. Therefore, in total we have 80 instances based on the Danish network and 80 instances based on the Dutch network.

6 Computational Results

We implemented the formulation presented in Section 4 and tested it on the 160 test instances described in 5 by solving it with CPlex 12.6 on a computer with an Intel CPU E5-2660 with a frequency of 2.60 GHz and equipped with 128 GB of RAM. All experiments have been conducted by using eight threads. We set a time limit of one hour on the solver for each run on each instance.

ID	$\lvert V^+\rvert$	$\lvert V^-\rvert$	$\overline{\delta}$	$\underline{tws}$	$\overline{twe}$	Avg CPU (s)	Avg Root Gap %	# Opt
1	3	3	10	18:00	02:00	16	1.50	5/5
2	3	5	10	18:00	02:00	48	0.12	5/5
3	5	3	10	18:00	02:00	263	3.76	5/5
4	5	5	10	18:00	02:00	141	3.40	5/5
5	3	3	25	18:00	02:00	15	0.78	5/5
6	3	5	25	18:00	02:00	41	0.65	5/5
7	5	3	25	18:00	02:00	60	3.58	5/5
8	5	5	25	18:00	02:00	165	5.75	5/5
9	3	3	10	13:00	02:00	17	0.97	5/5
10	3	5	10	13:00	02:00	50	0.56	5/5
11	5	3	10	13:00	02:00	58	6.00	5/5
12	5	5	10	13:00	02:00	115	0.76	5/5
13	3	3	25	13:00	02:00	20	4.56	5/5
14	3	5	25	13:00	02:00	43	1.00	5/5
15	5	3	25	13:00	02:00	57	16.56	5/5
16	5	5	25	13:00	02:00	110	0.56	5/5
						76	3.16	80/80

Table 1: Computational results on the 80 instances of the Danish network

In Table 1 and 2, we show the results obtained by solving the instances from the Danish and Dutch railway networks respectively. The instances in both cases have been created starting from different parameters values that are reported in the tables: number of surplus stations ($\lvert V^+\rvert$), number of deficits stations $\lvert V^-\rvert$), maximum dwell time ($\overline{\delta}$), and time windows ($\underline{tws}$, $\overline{twe}$). In both tables, each row reports the average results over the five instances having the same set of parameters; in particular, we report the instance block id (*ID*), the instance parameters (i.e., $\lvert V^+\rvert$, $\lvert V^-\rvert$, $\overline{\delta}$, $\underline{tws}$, $\overline{twe}$), the average computation time in seconds (*Avg. CPU Time (s)*), the average gap in percentage at the root node (*Avg. Root Gap %*), and the number of instances solved to optimality within the time limit (*# Opt.*). The last row of each table

ID	$\|V^+\|$	$\|V^-\|$	$\overline{\delta}$	$\underline{tws}$	$\overline{twe}$	Avg CPU (s)	Avg Root Gap %	# Opt
17	3	3	10	18:00	02:00	28	11.47	5/5
18	3	5	10	18:00	02:00	25	0.88	5/5
19	5	3	10	18:00	02:00	88	10.13	5/5
20	5	5	10	18:00	02:00	185	2.43	5/5
21	3	3	25	18:00	02:00	8	8.90	5/5
22	3	5	25	18:00	02:00	135	5.04	5/5
23	5	3	25	18:00	02:00	86	6.96	5/5
24	5	5	25	18:00	02:00	142	8.29	5/5
25	3	3	10	13:00	02:00	71	9.08	5/5
26	3	5	10	13:00	02:00	76	0.80	5/5
27	5	3	10	13:00	02:00	623	11.26	5/5
28	5	5	10	13:00	02:00	1144	25.60	4/5
29	3	3	25	13:00	02:00	28	9.22	5/5
30	3	5	25	13:00	02:00	534	4.54	5/5
31	5	3	25	13:00	02:00	1105	29.86	4/5
32	5	5	25	13:00	02:00	1215	12.80	5/5
						343	9.82	78/80

Table 2: Computational results on the 80 instances of the Danish network

shows the average computation time, average root gap, and the number of instances solved to optimality.

As Table 1 shows, the proposed MILP was able to solve all 80 instances from the Danish case within one hour of computation time with an average running time is 76 seconds. The instance block with the highest average running time is block 3 with an average of 263 seconds. For the Dutch case, as Table 2 shows, the proposed MILP was able to solve 78 instances within one hour of computation time and an average running time of 343 seconds. The instance block with the highest average running time is block 32 with an average of 1215 seconds. The average running times are in general higher respect to the Danish case, the lowest being eight seconds and the highest being about 20 minutes. This can depend on the structure of the two networks: as shown in Figures 3 and 4, the Dutch network is more complex and allows many more routes than the Danish network. We can also notice that the block of instances 25-32 have higher average running times compared to the block of instances 17-24; indeed, instance blocks 25-32 have larger time windows (from 13:00 to 02:00) and therefore more variables and constraints.

6.1 Lazy constraints

CPlex allows to set some of the constraints of a model as *lazy*. A lazy constraint is initially inactive and is added only if an achieved integer solution violates it. This dynamic management of lazy constraints can allow to reduce the overall computation time of the solver. In particular, it can be convenient to set as lazy all constraints that are more unlikely to be violated by the feasible solutions. We tested the behavior of the model when setting constraints (13)-(16) set as lazy and compare the results with the results previously reported. In Table 3 and 4, we compare the average computation time, the average gap at root node, and the

	Avg CPU (s)		Avg Root Gap %		# Opt	
ID	Original	Lazy	Original	Lazy	Original	Lazy
1	16	10	1.50	1.50	5/5	5/5
2	48	129	0.12	0.12	5/5	5/5
3	263	216	3.76	3.76	5/5	5/5
4	141	309	3.40	3.40	5/5	5/5
5	15	10	0.78	0.78	5/5	5/5
6	41	32	0.65	0.65	5/5	5/5
7	60	57	3.58	3.58	5/5	5/5
8	165	198	5.75	5.75	5/5	5/5
9	17	19	0.97	0.97	5/5	5/5
10	50	52	0.56	0.56	5/5	5/5
11	58	35	6.00	6.00	5/5	5/5
12	115	102	0.76	0.76	5/5	5/5
13	20	14	4.56	4.56	5/5	5/5
14	43	42	1.00	1.00	5/5	5/5
15	57	50	16.56	16.56	5/5	5/5
16	110	461	0.56	0.56	5/5	5/5
	76	108	3.16	3.16	80/80	80/80

Table 3: Impact of lazy constraints on instances from the Danish network

number of instances solved to optimality (within one hour) by the original formulation and the original formulation with lazy constraints.

The results obtained on the Danish instances (see Table 3) show that lazy constraints do not help decrease the computation time; indeed, it increases on average from 76 to 108 seconds. On the other hand in the Dutch cases (see Table 4), the total average running time decreases from 343 to 181. Moreover, by using lazy constraints, it was possible to solve an additional instance with the time limit. The different impact of using lazy constraints can be due to the structure of the two networks. In the Danish network, trains that cross the country west to east (or viceversa) are forced to go through the central part of the network where headway constraints can be particularly tight, so many more constraints (13)-(16) could be tight compared to the Dutch network where more routing options are available for each deadhead train.

6.2 Heuristic solution

We also implemented a fast and intuitive heuristic algorithm to solve the RSRP-DT and compared the solutions found with this heuristic with the solutions found with the MILP. Figure 5 illustrates how the algorithm works in a schematic way, and Algorithm 1 provides a pseudocode of the algorithm. The main idea of the heuristic algorithm is to solve one off-balance at a time by iteratively scheduling deadhead trains from the surplus station with the highest (remaining) surplus of train units to the deficit station with the highest (remaining) deficit of train units. At each iteration, for the selected pair of surplus station i and deficit station j, the algorithm tries to schedule a deadhead train from i to j that is compatible with time windows, with passenger train timetables, and that carries as many train units as possible

	Avg CPU (s)		Avg Root Gap %		# Opt	
ID	Original	Lazy	Original	Lazy	Original	Lazy
17	28	16	11.47	11.47	5/5	5/5
18	25	19	0.88	0.88	5/5	5/5
19	88	31	10.13	10.13	5/5	5/5
20	185	62	2.43	2.43	5/5	5/5
21	8	10	8.90	8.90	5/5	5/5
22	135	85	5.04	5.04	5/5	5/5
23	86	16	6.96	6.96	5/5	5/5
24	142	72	8.29	8.29	5/5	5/5
25	71	79	9.08	9.08	5/5	5/5
26	76	74	0.80	0.80	5/5	5/5
27	623	310	11.26	11.26	5/5	5/5
28	1144	482	25.60	25.60	4/5	5/5
29	28	30	9.22	9.22	5/5	5/5
30	534	140	4.54	4.54	5/5	5/5
31	1105	881	29.86	29.86	4/5	4/5
32	1215	600	12.80	12.80	5/5	5/5
	343	181	9.82	9.82	78/80	79/80

Table 4: Impact of lazy constraints on instances from the Dutch network

to satisfy either the surplus of i or the deficit of j or both. The process is repeated until either all surpluses/deficits are resolved (in this case, the RSRP-DT is considered *solved*) or by returning that it is not possible to solve all surpluses/deficits (in this case, the RSRP-DT is considered *unsolved*). Such an algorithm represents an intuitive way of heuristically solving the RSRP-DT.

Algorithm 1 Pseudocode of the heuristic to solve the RSRP-DT

while $V^+ \neq \emptyset$ **do** ▷ As long as there are unsolved surplus stations
  $i := getHighestSurplus(V^+)$ ▷ Retrieve station with highest surplus
  $V^+ := V^+ \setminus \{i\}$ ▷ Remove i from surplus stations
  $T := V^-$ ▷ Set T equal to the unresolved deficits
  while $a_i > 0$ and $T \neq \emptyset$ **do** ▷ As long as there are units at i and deficits to solve from i
    $j := getHighestDeficit(T)$ ▷ Retrieve station with highest deficit
    $T := T \setminus \{j\}$ ▷ Remove j from the deficits to solve from i
    $units := \min\{a_i, b_j\}$ ▷ Set the number of units to send from i to j
    $dhTrain := generateDeadheadTrain(i, j)$ ▷ Try to schedule a deadhead train from i to j
    if $dhTrain = Yes$ **then** ▷ If a deadhead train from i to j was scheduled
      $updateSolution(dhTrain)$ ▷ Add deadhead train to incumbent solution
      $a_i := a_i - units$ ▷ Update remaining surplus at i
      $b_j := b_j - units$ ▷ Update remaining deficit at j
      if $b_j = 0$ **then** ▷ If deficit j was solved
        $V^- := V^- \setminus \{j\}$ ▷ Remove j from unsolved deficits
if $V^- = \emptyset$ **then** ▷ If all off-balances where solved
  $output := solved$ ▷ Return $solved$
else
  $output := unsolved$ ▷ Otherwise return $unsolved$

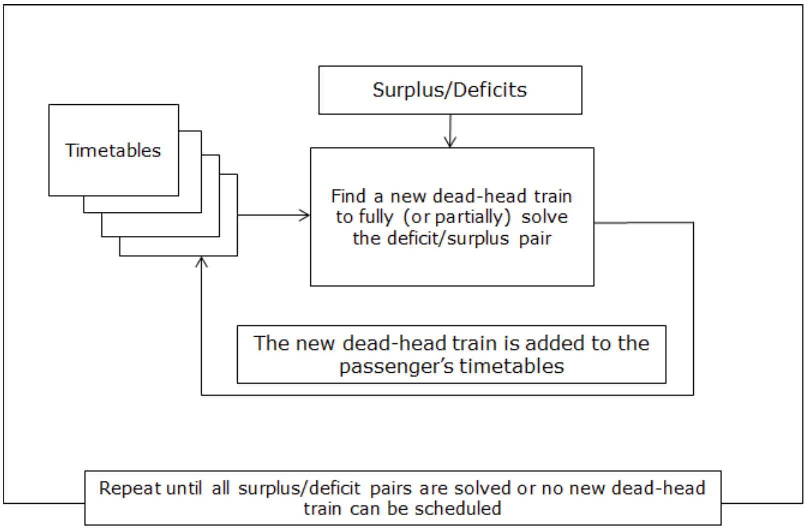

Fig. 5: Outline of the heuristic algorithm

	Avg CPU (s)			# Rebalanced	
ID	MILP	Heuristic	Avg Cost Increase %	MILP	Heuristic
1	16	2	9.8	5/5	5/5
2	48	1	27.3	5/5	5/5
3	263	1	6.6	5/5	5/5
4	141	1	23.7	5/5	5/5
5	15	0.5	9.5	5/5	5/5
6	41	0.5	2.0	5/5	5/5
7	60	0.5	21.1	5/5	5/5
8	165	0.5	35.2	5/5	5/5
9	17	0.5	18.5	5/5	5/5
10	50	1	7.9	5/5	5/5
11	58	1	30.8	5/5	5/5
12	115	0.5	51.1	5/5	5/5
13	20	1	26.2	5/5	5/5
14	43	1	9.7	5/5	5/5
15	57	1	25.4	5/5	5/5
16	110	1	32.2	5/5	5/5
	76	1	21.6	80/80	80/80

Table 5: Comparison between MILP and heuristic on the Danish network

Tables 5 and 6 provide a comparison between the results achieved by the MILP model and the heuristic algorithm on the Danish and the Dutch networks, respectively. For each block of instances, it is reported the average computation time of the MILP and the heuristic, the average cost increase in percentage of the solution returned by the heuristic compared to

	Avg CPU (s)			# Rebalanced	
ID	MILP	Heuristic	Avg Cost Increase %	MILP	Heuristic
17	28	1	9.1	5/5	5/5
18	25	1	22.3	5/5	5/5
19	88	0.5	23.7	5/5	5/5
20	185	1	30.5	5/5	5/5
21	8	1	20.1	5/5	5/5
22	135	0.5	50.3	5/5	3/5
23	86	1	10.6	5/5	5/5
24	142	1	17.3	5/5	5/5
25	71	1	9.1	5/5	5/5
26	76	1	20.2	5/5	5/5
27	623	1	29.4	5/5	5/5
28	1144	1	44.7	4/5	4/5
29	28	0.5	22.7	5/5	5/5
30	534	1	14.3	5/5	5/5
31	1105	1	18.8	4/5	4/5
32	1215	1	60.1	5/5	4/5
	343	1	25.2	78/80	75/80

Table 6: Comparison between MILP and heuristic on the Dutch network

the solution returned by the MILP, and the number of instances that were fully rebalanced by the two methods.

Both tables clearly show that the heuristic method is fast, but cost of the solutions returned is significantly higher than the cost of the solutions return by the MILP - on average, 21.6% higher on the Danish instances and 25.2% higher on the Dutch instances. Moreover, while all instances on the Danish network could be fully rebalanced by using the heuristic method, five of the 80 instances on the Dutch network could not be fully rebalanced.

7 Conclusion

This paper deals with the RSRP, which is a problem arising in rolling stock rescheduling when the rolling stock plan has been changed due to planned maintenance or unexpected events and results into off-balanced inventories of train units at certain stations of the railway network. We have investigated the solution of the RSRP by scheduling deadhead trains in between passenger trains to solve the off-balances by taking into account operational constraints, such as station capacities and headway times. We have presented a MILP model to formulate the RSRP and have test it on a set of real-life instances based on the Danish and the Dutch railway networks. The computational results have showed that the MILP model is able to solve an optimal set of deadhead trains to resolve the off-balances in most of the test instances within one hour of computation time. We have solved compared the results achieved by the MILP with the results achieved by an intuitive heuristic algorithm and have showed that the MILP can provide significantly less expensive solutions. The computational campaign conducted on real-life instances shows that the proposed MILP can be a valuable tool to use in practice to rebalance train unit inventories in practice.

This study can be of inspiration for further studies on the topic. In particular, we think it can be interesting to extend the model in three directions: (*i*) to accommodate different types of rolling stock units and compositions of heterogeneous rolling stock types, (*ii*) to integrate coupling and uncoupling operations, and (*iii*) to combine the scheduling of deadhead trains with changes in the rolling stock of the scheduled passenger trains.

References

N. Bešinović, R.M.P. Goverde, E. Quaglietta, and R. Roberti. An integrated micro-macro approach to robust railway timetabling. *Transportation Research Part B: Methodological*, 87:14–32, 2016.

R. Borndörfer, M. Grötschel, and M.E. Pfetsch. A column-generation approach to line planning in public transport. *Transportation Science*, 41(1):123–132, 2007.

G. Budai, G. Maróti, R. Dekker, D. Huisman, and L.G. Kroon. Rescheduling in passenger railways: The rolling stock rebalancing problem. *Journal of Scheduling*, 13(3):281–297, 2010.

V. Cacchiani, A. Caprara, L. Galli, L.G. Kroon, G. Maróti, and P. Toth. Railway rolling stock planning: Robustness against large disruptions. *Transportation Science*, 46(2):217–232, 2012.

V. Cacchiani, D. Huisman, M. Kidd, L.G. Kroon, P. Toth, L. Veelenturf, and J. Wagenaar. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, 63:15–37, 2014.

L. Cadarso, Á Marín, and G. Maróti. Recovery of disruptions in rapid transit networks. *Transportation Research Part E: Logistics and Transportation Review*, 53:15–33, 2013.

A. Caprara, M. Monaci, and P. Toth. *A Global Method for Crew Planning in Railway Applications*, pages 17–36. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

A. Caprara, M. Fischetti, and P. Toth. Modeling and solving the train timetabling problem. *Operations Research*, 50(5):851–861, 2002.

A. Caprara, L. Kroon, M. Monaci, M. Peeters, and P. Toth. Chapter 3 passenger railway optimization. In C. Barnhart and G. Laporte, editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, pages 129–187. Elsevier, 2007.

X. Delorme, J. Rodriguez, and X. Gandibleux. Heuristics for railway infrastructure saturation. *Electronic Notes in Theoretical Computer Science*, 50(1):39–53, 2001.

P.-J. Fioole, L.G. Kroon, G. Maróti, and A. Schrijver. A rolling stock circulation model for combining and splitting of passenger trains. *European Journal of Operational Research*, 174(2):1281–1297, 2006.

J.-W. Goossens, S. Van Hoesel, and L.G. Kroon. A branch-and-cut approach for solving railway line-planning problems. *Transportation Science*, 38(3):379–393, 2004.

J.T. Haahr, J.C. Wagenaar, L.P. Veelenturf, and L.G. Kroon. A comparison of two exact methods for passenger railway rolling stock (re)scheduling. *Transportation Research Part E: Logistics and Transportation Review*, 91:15–32, 2016.

D. Huisman, L.G. Kroon, R.M. Lentink, and M.J.C.M. Vromans. Operations research in passenger railway transportation. *Statistica Neerlandica*, 59(4):467–497, 2005.

J. Jespersen-Groth, D. Potthoff, J. Clausen, D. Huisman, L. Kroon, G. Maróti, and M.N. Nielsen. *Disruption Management in Passenger Railway Transportation*, pages 399–421. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

L.G. Kroon and M. Fischetti. Crew scheduling for Netherlands Railways "Destination: Customer". In *Computer-Aided Scheduling of Public Transport*, pages 181–201, 2001.

L.G. Kroon, G. Maróti, and L.K. Nielsen. Rescheduling of railway rolling stock with dynamic passenger flows. *Transportation Science*, 49(2):165–184, 2015.

C. Liebchen and R.H. Möhring. The modeling power of the periodic event scheduling problem: Railway timetables – and beyond. In F. Geraets, L. Kroon, A. Schoebel, D. Wagner, and C.D. Zaroliagis, editors, *Algorithmic Methods for Railway Optimization*, pages 3–40, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

R.M. Lusby, J.T. Haahr, J. Larsen, and D. Pisinger. A branch-and-price algorithm for railway rolling stock rescheduling. *Transportation Research Part B: Methodological*, 99:228–250, 2017.

L.K. Nielsen, L.G. Kroon, and G. Maróti. A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research*, 220(2): 496–509, 2012.

L. Peeters. *Cyclic railway timetable optimization*. PhD thesis, Erasmus Universiteit Rotterdam, 2003.

A. Schöbel. Line planning in public transportation: models and methods. *OR Spectrum*, 34 (3):491–510, 2012.

A. Schöbel and S. Scholl. Line planning with minimal traveling time. In L.G. Kroon and R.H. Möhring, editors, *5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS'05)*, volume 2, 2006.

A. Schrijver. Minimum circulation of railway stock. *CWI Quarterly*, 6(3):205–217, 1993.

L.P. Veelenturf, L.G. Kroon, and G. Maróti. Passenger oriented railway disruption management by adapting timetables and rolling stock schedules. *Transportation Research Part C: Emerging Technologies*, 80:133–147, 2017.

J.C. Wagenaar, L.G. Kroon, and I. Fragkos. Rolling stock rescheduling in passenger railway transportation using dead-heading trips and adjusted passenger demand. *Transportation Research Part B: Methodological*, 101:140–161, 2017.