



A Branch-and-Price algorithm for railway rolling stock rescheduling

Lusby, Richard Martin ; Haahr, Jørgen Thorlund; Larsen, Jesper; Pisinger, David

Published in:
Transportation Research. Part B: Methodological

Link to article, DOI:
[10.1016/j.trb.2017.03.003](https://doi.org/10.1016/j.trb.2017.03.003)

Publication date:
2017

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Lusby, R. M., Haahr, J. T., Larsen, J., & Pisinger, D. (2017). A Branch-and-Price algorithm for railway rolling stock rescheduling. *Transportation Research. Part B: Methodological*, 99, 228-250. DOI: 10.1016/j.trb.2017.03.003

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

A Branch-and-Price Algorithm for Railway Rolling Stock Rescheduling

Richard M. Lusby^{a,*}, Jørgen Thorlund Haahr^b, Jesper Larsen^a, David Pisinger^a

^a*Department of Engineering Management, Technical University of Denmark, Produktionstorvet, building 426, 2800 Kgs. Lyngby, Denmark*

^b*Optivation, Rued Langgaards Vej 7, 2300 Copenhagen, Denmark*

Abstract

How to best reschedule their fleet of rolling stock units during a disruption is an optimization problem regularly faced by railway operators. Despite the problem's high complexity, it is still usually solved manually. In this paper we propose a path based mathematical formulation and solve it using a Branch-and-Price algorithm. We demonstrate that, unlike flow based approaches, our formulation is more easily extended to handle certain families of constraints, such as train unit maintenance restrictions. The proposed algorithm is benchmarked on several real-life instances provided by the suburban railway operator in Copenhagen, DSB S-tog. When used in combination with a lower bound method taken from the literature we show that near-optimal solutions to this rescheduling problem can be found within a few seconds. Furthermore, we show that the proposed methodology can be used, with minor modification, on a tactical planning level, where it produces near-optimal rolling stock schedules in minutes of CPU time.

Keywords: Railway Optimization, Rolling Stock Rescheduling, Disruption Management

1. Introduction

During the daily operations of a [railway company](#) unplanned events inevitably occur, disrupting the planned service. Depending on the severity of a disruption, vehicle and crew schedules may need to be revised in order for the railway operator to continue to operate. Common examples of disruptions include, but are not limited to: crew sickness, vehicle malfunction, signalling system and infrastructure failures, and weather conditions. In some cases, such disruptions render the planned, schedules infeasible and require immediate response. Deciding on what actions to take during a disruption is termed *disruption management* and can be a non-trivial task (see, e.g., [19]). Ascertaining the extent of a disruption is not easy, and the large number of possible decisions makes it difficult for dispatchers to find high quality solutions to the rescheduling problems. If not handled well, the effects of a disruption can easily cascade, creating even more problems. In this paper we focus on rescheduling a fleet of rolling stock units during a disruption. This is termed the Rolling Stock Rescheduling Problem (RSRP); a problem which has received relatively limited attention in the literature. Some contributions do exist, see, e.g., [27, 20].

Given a disruption, the RSRP involves optimally reassigning a fleet of rolling stock units to a set of timetabled train services while respecting a number of operational constraints and business rules. Rolling stock units (henceforth referred to as units) are divided into a set of *types*. A unit refers to the smallest inseparable vehicle we consider, i.e., a set of rail cars which, for practical reasons, must remain connected together. Two units of the same type have the same physical characteristics. Units can be combined in different ways, yielding different so-called *compositions*. A composition is a set of units in a specific order. Different compositions give the

*Corresponding author

Email address: rm1u@dtu.dk (Richard M. Lusby)

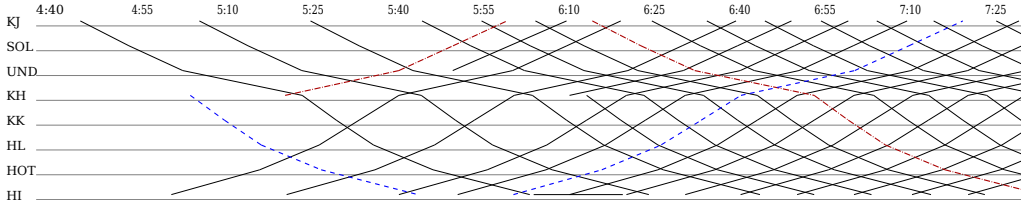


Figure 1: An example of a timetable for a single line with trip services going between two end stations. Two trip sequences marked red and dashed blue have been highlighted for illustrative purposes.

flexibility to modify the maximum number of passengers a service can cater for. Not all units can be assigned to all services, composition changes can only occur at certain stations, involving (de)coupling activities, whereby units must be removed from or added to a composition. Whenever (de)coupling is needed, units must be *shunted* from, or to, the station’s *depot*. A depot typically consists of several parallel tracks where units not in service can be parked. When solving the RSRP, consideration must be given to the available space in the entire network’s set of depots as well as the location where units terminate in the revised schedule. The aim is typically to cover the expected demand while minimizing operating cost and deviations from planned activities. From a rescheduling perspective, several feasible schedules may exist and it is important to find a cost-efficient solution as rolling stock is one of the major expenses for any railway company. Given the time pressure to quickly find a good solution during a disruption, we are motivated to develop an optimization based decision support system to solve the RSRP.

The main contributions of this paper are a new mathematical model for the RSRP and the development of a Branch-and-Price (BAP) algorithm to solve this formulation. Column generation is a well known technique that has been successfully applied to many problems. It is already state-of-the-art for railway crew rescheduling [29, 30]; however, applications in rolling stock scheduling are limited to tactical level planning problems (see, e.g., [25, 6, 8, 28]). To the best of our knowledge the proposed algorithm is the first attempt at using column generation for the RSRP. A key feature of the proposed mathematical model is that we model individual units as opposed to aggregated unit flows. Modelling on an individual unit level has the added benefit that unit specific constraints (e.g., maintenance restrictions) can be considered. The proposed methodology is benchmarked on real-life disruption instances provided by the suburban train operator in the Copenhagen region, DSB-Stog. We demonstrate that, when used in collaboration with a lower bound method taken from literature, near optimal rolling stock schedules are obtained within seconds. Additionally, we demonstrate the model’s ability to adhere to maintenance requirements as well as highlight the model’s applicability at a tactical level where it is used in a scheduling capacity as opposed to a rescheduling one. For the latter we show that optimized rolling stock schedules can be obtained in minutes.

This paper is structured as follows. A description of the problem is given in Section 2, and relevant literature is reviewed in Section 3. Section 4 presents our mathematical formulation of the RSRP, while Section 5 describes the BAP algorithm. Computational results are presented in 6, and the conclusions from this research are summarised in Section 7.

2. Problem Description

A timetable states the departure and arrival times of all operated train services. It usually consists of a set of *lines*, where each line typically runs between two end stations and stops at a sequence of intermediate stations. The term *trip* is used to refer to the traversal of a given line in a given direction at a certain time. In other words, a trip is an individual journey that starts at a particular station and terminates at a particular station (usually the line’s end stations). In some cases, a trip may originate or terminate at stations other than the corresponding line’s end stations. Figure 1 shows a time-space diagram for multiple trips of a specific line. A *trip sequence* denotes a continuous sequence of trips which can be performed by the same physical

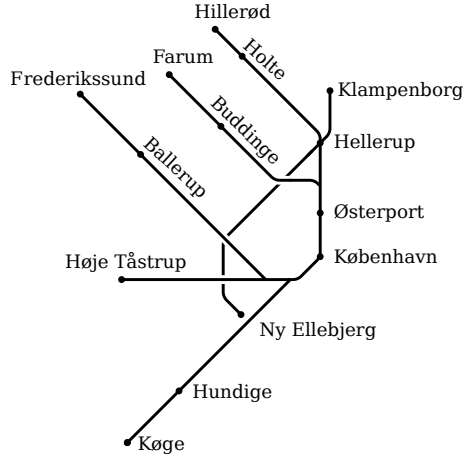


Figure 2: The S-tog railway infrastructure network. The depot and end stations are highlighted.

train composition. That is, if the same composition is assigned to every trip in the sequence then no units in the composition will be going in or out of a depot and no shunting activities are performed. Trip sequences typically span a whole day and stay within a single line. In some cases, a trip sequence alternates between two different lines in the network. [Note that rolling stock is allowed to circulate between several trip sequences.](#) The majority of trips are revenue generating; however, several *dead-heading* trips, or non-revenue generating train movements, may be scheduled to re-position units.

To provide a service, railway operators need a detailed schedule that specifies what rolling stock will be used between every two consecutive stops in the timetable and where any composition changes will occur. A trip can be partitioned into a sequence of *subtrips*, where each subtrip defines a shorter segment of the trip, which is delimited by key stations. Stations of interest could be, for example, terminal stations, turning stations, and stations where (de)coupling, and hence a composition change, is possible. Figure 2 illustrates such key stations in DSB S-tog’s network. Allowing composition changes from subtrip to subtrip provides more flexibility than simply fixing the composition of a train service for an entire trip.

Note that, depending on the type of disruption, there are potentially more such points of interest in the RSRP compared to its tactical level counterpart. An infrastructure blockage may require trains to turn around at stations where [they are](#) usually not permitted to do so.

At DSB S-tog, the fleet of units available is comprised of two different types, and all units can be coupled to any other unit in the fleet. The main distinguishing characteristics of each type are: the seating capacity, the unit’s length, and the operating cost. All units are self propelled, and all train drivers can operate both types. Examples of the various compositions are shown in Figure 3. In this work the compositions will be implicitly determined by the individual assignment of units to subtrips. We argue that the composition plays a smaller role in this study for two reasons. Firstly, compositions on subtrips typically range from one to two units in size at DSB S-tog. This means that there are no ordering conflicts when coupling or (de)coupling units. An uncommon composition with three units can occur; however, in this composition all units must be of the shorter type. Secondly, during a disruption, the ordering of individual units is not a governing priority.

In this paper, the definition of (de)coupling is consistent with the one which appears in several existing papers (see, e.g., [18]). Recall that a coupling is performed whenever a unit is moved from a depot and onto a train service. Likewise, a decoupling is performed whenever a unit leaves a train service and enters a depot. In both cases a change in train composition is made. Recall that a trip sequence can be serviced by the same composition without requiring any shunting activity; as long as a unit performs subtrips of the same trip sequence in succession, no coupling or decoupling is needed. A subtrip can have one predecessor and one successor subtrip which in

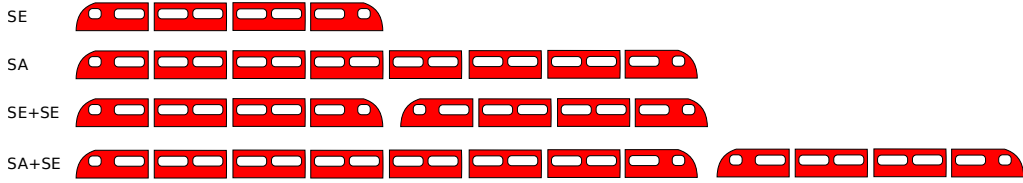


Figure 3: Examples of train compositions.

turn are in the same trip sequence.

Given a disruption, the RSRP entails reassigning units to every subtrip over a pre-specified time horizon. We assume a revised timetable is given, specifying which trips should be operated. Any changes to this compared to the original timetable may render the rolling stock schedules and other planned activities infeasible. Figure 4 provides an example of a disrupted line due to a blockage in both directions between two stations. In our case study the railway infrastructure owners have exclusive control over all train activities, while train operators (i.e., DSB S-tog) have a supporting role. The infrastructure owners respond to a disruption quickly by making trip cancellations or alterations. Predefined emergency plans exist and are used as guides for handling the disruption. Their immediate goal is to avoid delay propagation and further escalation. Once a revised timetable is available, a new candidate rolling stock schedule must be provided by the rolling stock dispatchers within a short time. This must adhere to a number of operational constraints such as fleet capacity, unit location, and depot capacity. Furthermore, each unit has a maximum distance it can travel before maintenance is necessary, known as a *distance-to-maintenance* restriction. Therefore a unit cannot be assigned a sequence of subtrips with total mileage greater than its distance-to-maintenance. Attempts are also made to operate as many subtrips as possible and to match the composition seating capacities with the forecast passenger demand on each subtrip. The obtained rolling stock schedule states a chronological list of tasks performed by every available unit.

In reality, the timetable and rolling stock schedule are not determined completely independently. The infrastructure owner cannot expect train operators to operate a timetable that is infeasible with respect to rolling stock, and they are also not interested in cancelling more train departures than necessary. Thus, when rescheduling rolling stock, it makes sense to integrate timetable decisions to some degree. In this paper we assume that subtrips do not have to be assigned any units; however, a large penalty cost is incurred every time this occurs. Not assigning any units to a subtrip indicates that it is cancelled. We do not, however, assume there is a penalty for cancelling deadheading subtrips as these do not affect the public timetable.

Depending on the time of the day and the location of the disruption, the available time before a decision must be made ranges from 3 to 20 minutes. Finding a feasible rolling stock schedule in such a short time frame is very challenging. The new plans may be far from optimal since the issues are usually resolved sequentially by degree of severity using rules of thumb, while only considering the immediate future. Units associated with cancelled trips are often put into the nearest depot without considering the long-term consequences. After the disruption is resolved the original timetable is restored by the infrastructure owners; at which point, DSB S-tog has, by contract, 60 minutes to recover the rolling stock (and all other) schedules. It may, however, be impossible to cover all subtrips at this point in time due to the effects of the disruption. Rolling stock dispatchers must thus be able to find new plans quickly both during and after the disruption. This motivates the use of a computer-aided decision support system.

The primary objective when solving the RSRP is feasibility; however, if more than one feasible schedule exists, some will be more attractive than others. Therefore, we also include the following secondary objectives: the number of subtrip cancellations, number of (de)couplings, seat shortage, end-of-day depot balances, and operating costs. Cancelling subtrips is highly undesirable as this is closely connected to DSB S-tog’s reliability performance. Although perhaps necessary at times, (de)coupling activities may affect the robustness of a schedule. A (de)coupling can easily take more time than is expected due to manual interaction and unforeseen technical difficulties. In

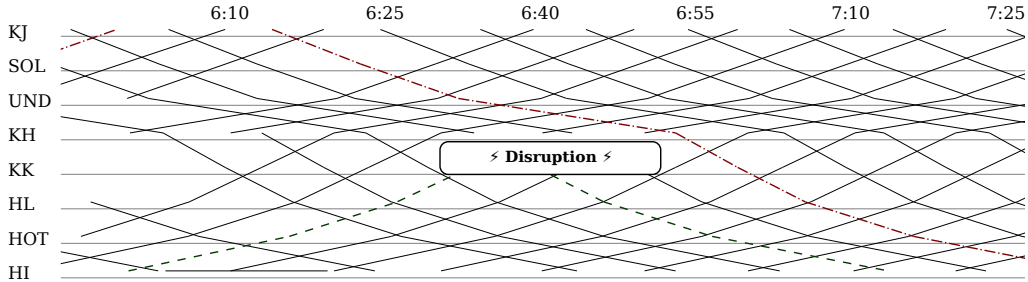


Figure 4: A disrupted timetable due to infrastructure blockage between København Central Station (KH) and Østerport Station (KK). The updated timetable resolves the situation by turning the train services at stations KH and KK.

addition, it also requires coordination with a second train or a depot driver. Minimizing the number of (de)couplings is thus important. Performing (de)couplings outside critical time periods as well as reserving enough time for the required operations could also be beneficial. In this paper we impose a minimum time required to perform (de)couplings. Minimizing seat shortage is also important. The train operator attempts to satisfy the actual demand on a trip by supplying enough seats to cover some percentage of the passengers. Therefore, it is not only necessary to cover a trip but also to assign units (i.e., a composition) that collectively provide enough seats. The end-of-day balance specifies how many units of each type are required in each depot at the end of the day to ensure operations can start smoothly the following morning. It is therefore important, when recovering a schedule, that this is adhered to as closely as possible. Otherwise, expensive deadheading movements will be required to reposition the units. Finally, it is important that unnecessary use of the units is avoided as units are expensive to operate.

In the literature, and also at DSB S-tog, the number of cancellations is usually the most important objective and is therefore given a cost that is significantly higher than the other objectives. The seat and operational objectives are conflicting and a non-trivial trade-off must be found. A train operator strives to meet the passenger demand, but not at all costs. For example, the operator would not couple an extra unit on a subtrip to ensure one extra seat. In the computational tests in Section 6, we investigate this trade-off and try to find a sensible balance. In rescheduling problems it seems natural to minimize the deviation from the original schedule. Examples include [27] and [30]. In this regard, we penalize the end-of-day imbalances, but do not penalize missed or unplanned shunting operations (in addition to the normal cost) during disruptions. Changing plans is understandably a nuisance that requires communication and coordination. However, with the recent advancements in mobile technology we believe that this is a diminishing issue.

3. Literature

Most literature related to passenger railway optimization (from an operator perspective) can be categorized by the problem it addresses, e.g., timetable planning, rolling stock scheduling, crew scheduling, or shunting. In this section we limit our discussion to the somewhat more scarce literature on rolling stock scheduling with an emphasis on real-time disruption management. For an overview of different railway optimization problems the reader is referred to [13], while [10], [24], and [21] provide overviews in a disruption management context. We consider self propelled trains units that do not require a powered locomotive for pulling the train. Several papers exist that consider the locomotive and carriages problem, see, e.g., [1, 14, 15, 16, 33, 26].

The rolling stock scheduling problem has been studied in a variety of different settings using different mathematical models and targeted time horizons. The rolling stock scheduling problem faced by the Dutch railway company Nederlandse Spoorwegen (NS), described in [18], is a good reference for the problem we consider. The problem involves assigning compositions to every departure. The aim is to cover every train service while respecting several constraints such as fleet capacity, composition change restrictions and allowing trains to be combined and split.

The strategic or tactical rolling stock scheduling problem has been studied in various forms over the past two decades. The aim of such problems is to determine the fleet size or to find a feasible rolling stock schedule. An early approach by Schrijver [32] minimizes the number of required trains to satisfy passenger demand. This approach does not consider train compositions, composition changes, depot parking nor maintenance. A small data-instance from NS is presented and solved within seconds.

A richer formulation is considered in [3]. The model considers the compositions of train arrivals and departures, assuming a single line and one family of compatible units. A generic workday on a line from NS is solved within 1-2 hours. We essentially consider a variant of this rolling stock scheduling problem; however, we also model individual rolling stock trajectories and restrict the number of parked trains to the capacity, i.e., total track length in meters, at each station depot. We do not include constraints limiting the composition changes.

A similar setting spanning multiple lines using one family of unit types is the focus of [28]. The authors present a BAP method which outperforms a commercial solver. Computation times are less than a minute for the larger data instance consisting of three lines. A few what-if scenarios are considered that quantify the effect of changing the fleet composition (and platform lengths) and allow shunting at a new station. Compared to our work, the authors present a different column generation decomposition of the formulation in [3]; sequences of composition changes are modelled as variables, while we model unit trajectories as variables.

An extension of the model in [3] is described in [18]. This model allows combining and splitting trains. The authors consider a subset of lines from NS and solve the problem as a Mixed Integer Program (MIP) using a commercial solver. In addition, a heuristic based on the Linear Program (LP) relaxation is used. Good quality solutions are found within hours. The solution model was used as a supportive tool for generating the Dutch timetable of 2005.

In [11], the authors consider a formulation similar to [18], [trying](#) to optimize the robustness of the rolling stock schedule while penalizing high densities of standing passengers. Deadheading units is allowed in order to increase capacity. Robustness is modelled by penalizing the total number of coupling operations, and by giving shunting operations during rush hours a special penalty. Two data instances are considered: a single day of a single line from the Spanish rail company, RENFE, with 4 depot stations and 320 train services, and two lines with 9 depot stations and 400 train services. Optimality is proven within minutes using a commercial MIP solver.

A two-stage optimization model is presented in [9] for solving the rolling stock scheduling problem with a set of generated failure scenarios. The model is similar to the one presented in [18], but without minimum shunting time, nor the combination and split of trains. A full MIP is presented and solved heuristically using Benders decomposition. The data instance considers one day of single line from NS with up to 400 trips and 28 failure scenarios. The solution method is able to find good solutions within 2 minutes.

Assigning self propelled units is the focus of [8]. The authors propose a heuristic based on a so-called customary column generation methodology. Solely the number of units used is minimized. [The paper also presents](#) several theorems and one lower bound scheme. Unlike our approach, [8] do not include costs for shunting, seat shortage and driven mileage. Furthermore, demand is modelled as a hard constraint and shunting capacity constraints at station depots are not included. An extension considering maintenance is proposed stating that all units must contain at least one maintenance opportunity during a weekly schedule.

A column generation approach for the Train Unit Scheduling Problem is presented in [25]. The authors consider the tactical level problem of generating a rolling stock schedule, propose an integer multicommodity flow formulation not that dissimilar to the formulation we present, and describe a BAP procedure to solve it. Promising results are obtained on real life instances that are provided [by](#) the major passenger train operator in Scotland, ScotRail. The largest of these instances contains around 500 train trips. As a comparison, we consider [instances with more than](#) 1,000 trips.

A locomotive assignment problem, which is structurally similar to the problem considered in this paper, is presented in [33]. The authors propose a column generation solution method that relies on heuristic fixing in a Branch-and-Bound (BAB) methodology. Sufficient engine capacity

(number of locomotives, horsepower and tonnage) must be assigned to train segments in order to pull the preassigned train cars. Like our work, units must be assigned to cover the train segments in order to meet a demand and total mileage is penalized. In contrast, we propose a full branching scheme and explicitly model train (de)coupling activities. Other minor modeling differences exist such as allowing cancellations, and end-of-day balance deviations.

The assignment of locomotives and cars at VIA Rail Canada is discussed in [15]. In contrast to our work, this problem involves assigning rail cars to satisfy demand as well as sufficient traction engines to pull the cars. The authors benchmark a column generation framework using heuristic branching and fixing rules to obtain solutions of good quality. In comparison, we model seat shortages and cancellations as soft constraints using penalties and do not need constraints to ensure sufficient traction. We include coupling costs directly in our algorithm, whereas [15] resolve these switching operations in a second phase. Depot capacity constraints are modelled, and sufficient time in a single planning cycle to perform maintenance is ensured. Subsequent work in [16] presents a benders decomposition framework that improves computational times, especially for the larger instances.

Several studies consider a short-term rolling stock problem aiming at rescheduling a few days before the day of operation. Compared to a long-term model such approaches have to deal with a greater level of detail and cannot assume a clean-slate approach as the availability of the different resources is fixed. In [7] a rebalancing problem, where the goal is to minimize inventory end-of-day deviations from the original plan, is considered. Two very fast heuristic methods, which aim to improve existing schedules, are presented. The authors prove that the rebalancing problem is NP-hard. The problem of rescheduling locomotives and carriages at VIA Rail Canada is studied in [26], while in [5], a schedule planning system and a tactical capacity-adjustment system for increasing revenue at SNCF is described.

A version of the RSRP is considered in [27]. The authors address a problem that is very similar to the one that we consider. A rolling stock scheduling framework is presented that uses time horizons to overcome disruption uncertainty and runtime difficulty. The optimization model is similar to the model presented in [18], but adapted to be executed in a rolling horizon. The main goal is to minimize the number of cancelled trains, end-of-day off-balances and changes to the original shunting plan. The end-of-day inventory balances are heuristically determined through the shorter horizons; each of which is solved using a commercial MIP solver.

Recovering rolling stock schedules is also the focus of [12]. The authors present an approach which integrates rolling stock and dynamic passenger demand. The model may cancel train services on predetermined lines, insert emergency train services and determine the direction of one-way tracks in case of one-way blocked segment. Passenger demand is updated iteratively after solving the rolling stock model until the demand stabilizes. The solution method is benchmarked on a 2 hour disruption case study provided by RENFE, consisting of 5 lines, 10 depot stations and 760 train services. Solutions are found within a few minutes. The case study does not reveal significant changes in passenger demand after the first iteration.

Finally, a variety of DSB S-tog related problems are studied in [19]. One of these focuses on the reinsertion of train lines once a disruption has been resolved. The model considers the distribution of available trains and the time required for drivers to reach them. The model is solved as a MIP which minimizes the time of the last inserted train. Test instances are solved within 30 seconds.

In summary, various models and methods have been proposed to solve variants of the rolling stock scheduling problem (both in planning and from a disruption perspective). To the best of the authors' knowledge, this paper is the first to present a path based formulation of the RSRP. We describe a BAP approach to solve this formulation, highlight its flexibility at including certain maintenance restrictions, and also demonstrate its applicability to the tactical level planning problem of rolling stock scheduling.

4. Model

We now define the RSRP and our notation more formally before presenting the proposed mathematical model. To distinguish between parameters and decision variables, the latter always

\mathcal{S}	Set of stations
\mathcal{D}	Set of stations that have an associated depot
\mathcal{T}	Set of subtrips
\mathcal{U}	Set of unit types
\mathcal{A}	Set of all arrival events across all stations in \mathcal{S}
\mathcal{P}_d	Set of all possible paths starting at station d
\mathcal{P}	Set of all possible paths

Table 1: List of sets

λ_p^u	An integer variable which determines how many units of type u travel on path $p \in \mathcal{P}$
y_t	Binary slack variable which takes value 1 iff subtrip $t \in \mathcal{T}$ is not covered
z_t	Integer slack variable determining the seat shortage on subtrip $t \in \mathcal{T}$
w_d^u	Integer slack variable determining end-of-day balance shortage for unit type $u \in \mathcal{U}$ at depot $d \in \mathcal{D}$

Table 2: List of Variables

appear in bold type. We denote by \mathcal{S} the set of key stations, which is an appropriate aggregation of all stations in the railway infrastructure. The set $\mathcal{D} \subseteq \mathcal{S}$ defines the set of stations with an associated depot. Furthermore, we let \mathcal{U} be the set of all unit types. Finally, we let \mathcal{T} be the set of all timetabled subtrips that have to be serviced. A subtrip $t \in \mathcal{T}$ departs at a certain time from a source station and arrives at a certain time at a target station. For convenience, we define \mathcal{A} to be the set of all subtrip arrivals. We define $depot(a) \in \mathcal{D}$ as the shorthand for the depot of the station where the arrival event $a \in \mathcal{A}$ occurs.

The RSRP is the problem of finding an assignment of all available units to subtrips. The assignment of each unit must respect the inherent temporal and spatial constraints. In other words, a unit can only be in one place at a time and can only be assigned to a subtrip if it is located at the departure station of the subtrip prior to the subtrip’s departure time. A unit entering or leaving a subtrip must have reserved enough time to perform the (de)coupling before being assigned to any other activity. Finally, the depot parking space is limited. This means that only a certain total length of units can stay parked in depots at any point in time. Consequently, a unit cannot perform a subtrip and then park if the destination depot is full.

Trip coverage is directly related to trip cancellations, and it is our primary concern to minimize the number of cancellations. If a subtrip in $t \in \mathcal{T}$ is uncovered, the trip is considered to be cancelled. Matching the demand is also important; thus, the seat shortage is minimized. The number of available seats depends on how many units service a subtrip. The difference between the available seats and the demand defines the shortage. However, if there is a surplus of seats then the shortage is zero. The number of (de)couplings affects the robustness of the solution, and hence should be minimized. We approximate the number of (de)couplings by counting the number of times a unit enters and leaves a subtrip. This is not an accurate number of required shunting operations; however, it measures how many times we change the composition of a trip which is directly related to robustness.

We formulate the RSRP using a path based model with an exponential number of path variables. We let \mathcal{P}_d denote the set of all paths originating at depot $d \in \mathcal{D}$. A path defines a feasible trajectory of a unit and indicates a sequence of subtrips which are serviced by a unit that is following the path. We only consider paths that can be feasibly performed with respect to time and spacial constraints. For example, a path cannot contain two subtrips that overlap in time nor can it perform two tasks in sequence if the arrival station of the first is not equal to the departure station of the second. The cost of a path depends on the sum of accumulated mileage and number of (de)couplings performed. Since a path is just a sequence of subtrips, it is easy to count the

c_1^t	Cost of cancelling a subtrip $t \in \mathcal{T}$
c_2	Cost of one seat shortage per kilometre
c_3	Cost of one end-of-day balance shortage
c_4	Cost of one coupling or decoupling
c_5^u	Operational cost per kilometre for unit type $u \in \mathcal{U}$
s_u	Number of seats on train unit $u \in \mathcal{U}$
l_u	Length of train unit $u \in \mathcal{U}$ in metres
$demand_t$	Seat demand for subtrip $t \in \mathcal{T}$
$length_t$	Maximum train composition length allowed for subtrip $t \in \mathcal{T}$
$inventory_d^u$	Number of units of type $u \in \mathcal{U}$ starting at depot $d \in \mathcal{D}$
eod_d^u	Target end-of-day balance for train unit $u \in \mathcal{U}$ at depot $d \in \mathcal{D}$
$track_d$	Combined length of tracks at depot $d \in \mathcal{D}$

Table 3: List of parameters

κ_t	Number of kilometres on subtrip $t \in \mathcal{T}$
ξ_p	Number of couplings and decouplings imposed by path $p \in \mathcal{P}$
κ_p	Number of travel kilometres accumulated on path $p \in \mathcal{P}$
α_p^t	Binary coefficient which takes the value 1 iff path $p \in \mathcal{P}$ visits subtrip $t \in \mathcal{T}$
β_p^d	Binary coefficient which takes the value 1 iff path $p \in \mathcal{P}$ terminates at depot $d \in \mathcal{D}$
$\gamma_p^{d,a}$	Binary coefficient which takes the value 1 iff path $p \in \mathcal{P}$ is staying at depot $d \in \mathcal{D}$ on or before the arrival of subtrip $a \in \mathcal{A}$

Table 4: List of Coefficients

number of times a unit following the path would enter or exit a depot; we assume a unit has to be parked whenever the dwell time between two consecutive subtrips is above a certain threshold (i.e., 10 minutes). The RSRP involves assigning exactly one path to each unit. Note that the proposed approach does not distinguish between units in service and idle units when a disruption occurs; both are rescheduled in exactly the same way. The only difference is that an idle unit may be rescheduled immediately, while a unit in service can only be reassigned a different sequence of subtrips once it has arrived at the next depot in its planned itinerary.

The mathematical model contains four sets of decision variables. First, $\lambda_p^u \in \mathbb{Z}_0^+$ controls how many units of type $u \in \mathcal{U}$ use path $p \in \mathcal{P}$, where $\mathcal{P} := \bigcup_{d \in \mathcal{D}} \mathcal{P}_d$. Second, $\mathbf{y}_t \in \{0, 1\}$ is a slack variable that determines whether a subtrip $t \in \mathcal{T}$ is cancelled. If set to value 1 subtrip t is cancelled, otherwise the variable has value 0 and t is not cancelled. Third, $\mathbf{z}_t \in \mathbb{Z}_0^+$ is a slack variable that counts the total seat shortage on subtrip $t \in \mathcal{T}$. Finally, $\mathbf{w}_d^u \in \mathbb{Z}_0^+$ denotes the number of units of type $u \in \mathcal{U}$ that are missing in depot $d \in \mathcal{D}$ from the scheduled end-of-day balance. The model now becomes.

$$\text{Minimize: } \sum_{t \in \mathcal{T}} (c_1^t \mathbf{y}_t + c_2 \kappa_t \mathbf{z}_t) \quad (1)$$

$$+ \sum_{u \in \mathcal{U}} \sum_{d \in \mathcal{D}} c_3 \mathbf{w}_d^u \quad (2)$$

$$+ \sum_{u \in \mathcal{U}} \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} (c_4 \xi_p + c_5^u \kappa_p) \lambda_p^u \quad (3)$$

The notation is summarised in Tables 1, 2, 3 and 4. The objective function is a weighted sum of five components. In (1) the cost of cancellations and seat-shortage are added. In (2) the end-of-day balance shortage cost is added. Finally, in (3) the (de)coupling and mileage costs are added.

$$\sum_{p \in \mathcal{P}_d} \lambda_p^u = \text{inventory}_d^u \quad (\pi_d^u) \quad \forall u \in \mathcal{U}, d \in \mathcal{D} \quad (4)$$

$$\sum_{u \in \mathcal{U}} \sum_{p \in \mathcal{P}} \alpha_p^t \lambda_p^u \geq 1 - \mathbf{y}_t \quad (\mu_t) \quad \forall t \in \mathcal{T} \quad (5)$$

$$\sum_{u \in \mathcal{U}} s_u \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} \alpha_p^t \lambda_p^u \geq \text{demand}_t - \mathbf{z}_t \quad (\delta_t) \quad \forall t \in \mathcal{T} \quad (6)$$

$$\sum_{p \in \mathcal{P}} \beta_p^d \lambda_p^u \geq \text{eod}_d^u - \mathbf{w}_d^u \quad (\omega_d^u) \quad \forall u \in \mathcal{U}, d \in \mathcal{D} \quad (7)$$

$$\sum_{u \in \mathcal{U}} l_u \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} \alpha_p^t \lambda_p^u \leq \text{length}_t \quad (\phi_t) \quad \forall t \in \mathcal{T} \quad (8)$$

$$\sum_{u \in \mathcal{U}} l_u \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} \gamma_p^{\text{depot}(a), a} \lambda_p^u \leq \text{track}_{\text{depot}(a)} \quad (\nu_a) \quad \forall a \in \mathcal{A} \quad (9)$$

$$\mathbf{y}_t \in \{0, 1\} \quad \forall t \in \mathcal{T} \quad (10)$$

$$\lambda_p^u \in \mathbb{Z}_0^+ \quad \forall p \in \mathcal{P}, u \in \mathcal{U} \quad (11)$$

$$\mathbf{z}_t \in \mathbb{Z}_0^+ \quad \forall t \in \mathcal{T} \quad (12)$$

$$\mathbf{w}_d^u \in \mathbb{Z}_0^+ \quad \forall u \in \mathcal{U}, d \in \mathcal{D} \quad (13)$$

Constraints (4) ensure that the number of paths for a specific unit type leaving a depot corresponds to the number of units of that type available at the depot. Note that these constraints must hold with equality as the subproblem network includes the possibility of assigning empty, or subtrip-less, paths. Constraints (5) **make sure** that \mathbf{y}_t is set to 1 if subtrip t is not covered by any path. Constraints (6) ensure that the lack of seats is captured by \mathbf{z}_t , while Constraints (7) make sure that the end-of-day balance shortage is captured in \mathbf{w}_d^u for every depot d and unit type u . Note, that we could as easily model and penalize surpluses; however, assuming the total end-of-day balances equals the fleet size one rolling stock shortage necessarily results in one surplus. Thus, it is unnecessary to penalize both. The maximum train composition length is restricted by (8), according to the platform lengths at the subtrip's endpoints. Constraints (9) enforce the requirement that the capacity at any depot d must be respected by every arrival event. If the track capacity at every arrival is satisfied up until the last arrival, then the schedule respects the depot capacities throughout the day. Paths can go in and out of multiple depots multiple times during the horizon, and multiplying $\gamma_p^{d,a}$ by the number of units currently residing in the depot provides the total track usage. Finally, Constraints (10)-(13) restrict the domains of the variables. Note that the slack variables are naturally integral if the path variables are integral. The notation for the dual values associated with the constraints of the model's LP relaxation are given in parentheses. An important observation is that when including maintenance requirements, the structure of Model (1)–(13) remains unchanged, since the constraints can be implicitly handled in the construction of the columns. In other words, a column corresponds to a legal sequence of subtrips for a specific unit type. Given the exponential number of path variables, it is impractical, **and in certain cases** even impossible, to generate them all a priori and explicitly include them in the formulation. Instead we propose to solve the LP relaxation of Model (1)–(13) using delayed column generation in which favorable unit paths are generated dynamically. We discuss this in detail in Section 5.

We conclude this section with a brief discussion on the modifications required in order for it to use Model (1)–(13) for the purposed of solving the rolling stock scheduling problem on a tactical planning level. The main differences essentially concern the necessity of constraints (4) and (7), as well as the nature of constraints (5). Typically, initial inventories and end of day balances are not fixed when planning the circulation of rolling stock units. We, however, assume that the planner will experiment with different scenarios or try to minimize the needed inventory. Furthermore, it is usually not common to allow cancellations at the planning level. While this

can be more strictly enforced by removing the \mathbf{y}_t variables, a sufficiently high penalty in the objective for cancellations will achieve the desired result. Arguably, the input and model required for solving the tactical and rescheduling case is very similar. In Section 6 we investigate different planning scenarios by considering different parameter settings for the different costs. In particular we examine how the algorithm behaves when 1) including cancellation costs and a small cost for coupling; 2) introducing a seat shortage penalty; 3) introducing a mileage penalty; and 4) introducing maintenance constraints.

5. A Branch-and-Price Algorithm

In this section we propose a BAP algorithm for solving the RSRP. This well known technique for solving large-scale integer programs combines column generation with BAB [4]. Column generation is typically preferred when the mathematical model contains a vast number of variables. To implement column generation, one relaxes all integrality restrictions and decomposes the problem into a *master problem* and one or more independent *subproblems*. The master problem contains only a subset of the variables for the full problem, and is typically referred to as the Restricted Relaxed Master Problem (RRMP). The subproblems are optimization problems responsible for generating variables (or columns) that are not included in the RRMP, but which have the potential to decrease the RRMP’s objective function value. More specifically, each subproblem utilizes the dual information from an optimal solution to the RRMP, and attempts to identify negative reduced cost columns that can be added to the RRMP. Column generation refers to the iterative procedure between master and subproblems that must be performed. Incorporating this into a BAB setting in which columns are generated (or “priced”) at each node of the BAB tree gives rise to the BAP terminology. This section focuses on specific details concerning our BAP approach.

Figure 5 illustrates the steps in the algorithm. First, a master problem is generated with an initial subset of columns. This set of columns can be obtained using a simple greedy heuristic that ensures a feasible solution to the RRMP exists. Different greedy strategies were explored; however, preliminary testing showed no significant improvement over just initially assigning an empty itinerary to each of the rolling stock units. The RRMP is solved and the dual values are used to find variables with negative reduced cost by the subproblem. In this implementation of BAP, when solving a subproblem we find, and use, if favourable, the lowest reduced cost path per unit type. Whenever new variables are found the RRMP must be re-solved, otherwise the RRMP is optimal. A BAB node is finished when an optimal solution to the RRMP is found. If the solution is feasible and respects the integral constraints of the original master problem, then the upper bound is updated otherwise new branching nodes are created. The algorithm terminates once all nodes are processed.

5.1. Master Problem

The RRMP is obtained from Model (1)-(13) by replacing Constraints (10)-(13) with

$$\mathbf{y}_t \in [0, 1] \quad \forall t \in \mathcal{T} \quad (14)$$

$$\lambda_p^u \in \mathbb{R}_0^+ \quad \forall p \in \mathcal{P}, \forall u \in \mathcal{U} \quad (15)$$

$$\mathbf{z}_t \in \mathbb{R}_0^+ \quad \forall t \in \mathcal{T} \quad (16)$$

$$\mathbf{w}_d^u \in \mathbb{R}_0^+ \quad \forall u \in \mathcal{U}, d \in \mathcal{D} \quad (17)$$

and restricting the set of path variables from depot $d \in \mathcal{D}$ to a subset of paths $\mathbb{P}_d \subset \mathcal{P}_d$. Each column defines a legal sequence of subtrips for a unit originating from depot d .

5.2. Subproblem

The role of a subproblem is to identify one or more negative reduced cost columns. As indicated above, this can be viewed as the pricing step in the conventional simplex algorithm, with the exception that all variables are not stored explicitly. Variables with negative reduced cost for the

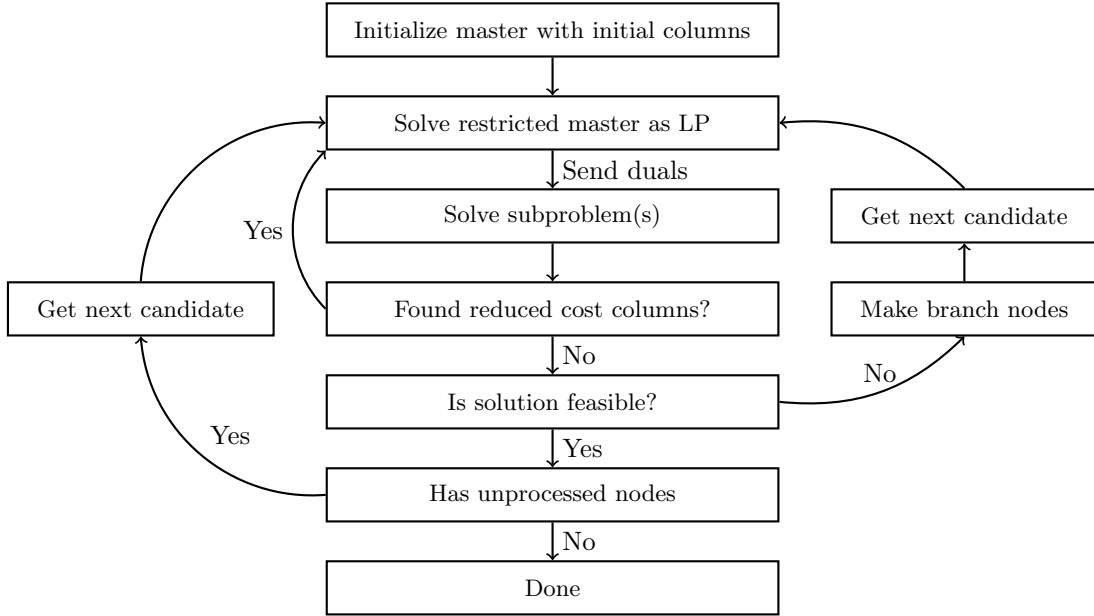


Figure 5: Flow diagram for the proposed Branch-And-Price framework

RRMP are returned by the subproblem(s). We formulate the problem of finding such variables as a Shortest Path Problem on an acyclic time-space network (see, e.g., [33] and [15]). A subproblem is defined for each unit type. This implies that, in the absence of individual restrictions, any unit of the given type can perform the path. When individual restrictions, e.g., maintenance are considered, a unit can only perform a path if sufficient miles (or time) is left before the unit requires service. As mentioned above, with or without such constraints, the master problem remains unchanged, this extension only affects the subproblem. The underlying network structure of each subproblem is identical since we assume any unit can perform any subtrip; however, some parameters of each subproblem must be modified to reflect the individual unit types. In what follows we present a more detailed description of the time-space network. To ease the understanding, the network is introduced in stages.

5.2.1. Underlying Network

To generate a column, one must identify a feasible sequence of subtrips starting at one depot and ending at another. Modelling all such possibilities for a given planning horizon can be achieved via a time-space network. In such a network, every node corresponds to a particular *event*. Here an event can refer to a departure, a passthrough, or an arrival. A departure event refers to the start of a trip sequence, while an arrival event is associated with the end of a trip sequence. A passthrough event represents a midway stop between two subtrips and has one incoming and one outgoing subtrip. Note that, in implementation, such events should be preprocessed as two such subtrips can be reduced to one. As every subtrip indicates the movement of a train between two particular stations at a certain time, it can be introduced as an edge in the time-space network.

Associated with each subtrip edge is a passenger demand and an operating cost $c_5^u \kappa_t$. Recall that the operating cost depends on the unit type under consideration. Furthermore, each subtrip is associated with a constraint from (5), a constraint from (6), and a constraint from (8). Thus, the reduced cost contribution (or weight) of each such edge is $\rho_t := c_5^u \kappa_t - \mu_t - s_u \delta_t - l_u \phi_t$. Additionally, edges which indicate a depot arrival are adjusted by $-l_u \nu_a$ consumption of depot track capacity (see Constraints (9)). By adding a super source as well as a super sink node, connecting the former to all departure events and the latter to all arrival events, we obtain an acyclic, time-space network, in which all trips within the planning period are implicitly represented by paths. Note that the weight of edges emanating the origin is $c_4 - \pi_d^u$, while edges to the sink have weight

$c_4 - \omega_d^u$. The cost c_4 indicates that the unit enters or exits a trip sequence.

Figure 6 below illustrates such a time-space network for a simplified example with three stations, two of which are depot stations (denoted d_1 and d_2). The example contains two trip sequences (each having three trips) having 6 subtrips each (numbered 1-6 and 7-12, respectively). All nodes corresponding to station events are ordered in time, and appear on the same horizontal level. The network is labelled assuming we are pricing a unit of type $u \in \mathcal{U}$. An example path from depot station $d_1 \in \mathcal{D}$ to depot $d_2 \in \mathcal{D}$ is given, along with a selection of example edge weights. Note that this network does not allow (de)coupling during a trip. The subtrips $a \in \mathcal{A}$ and $a' \in \mathcal{A}$ denote the two arriving subtrips, one at each depot (subtrips 12 and 6, respectively).

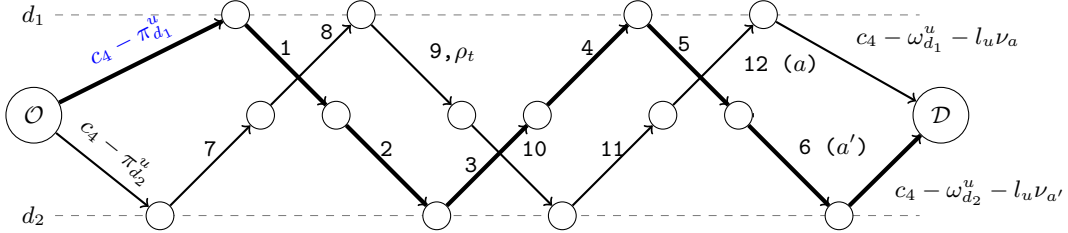


Figure 6: Simplified example of the acyclic time-space network. This figure illustrates the subtrips of two trip sequences. The black solid path is an example of a possible unit path. The current network does not capture depot details.

While this time-space network includes all subtrips in the planning horizon, it is inflexible in the sense that it does not permit a unit to be decoupled from a trip sequence; since each node in the graph has degree at most two, each path must consist of a sequence of consecutive subtrips. This is not ideal; it may be preferable to allow units to follow only part of a trip sequence. To provide such flexibility, we duplicate all nodes for station events occurring at depot-stations, creating a *platform event* node as well as a *depot event* node. This creates more possibilities for a unit, given its location at the time of the respective station event. A platform event stipulates that the unit type is at the platform and will perform the outgoing subtrip, while a depot event indicates that the unit is positioned in the depot and cannot perform the outgoing subtrip.

To allow a unit type to remain parked at a depot-station we introduce edges connecting consecutive depot events at the same station. Similarly, we introduce *coupling edges*. Such edges connect depot events with platform events and indicate that the unit type will be coupled to other units to perform the outgoing subtrip. Coupling edges must respect a minimum coupling time, meaning a depot event cannot connect to the station platform event before the coupling time has elapsed. Similarly, *decoupling edges* are introduced, each of which connects a platform event with a depot event. In contrast to coupling edges, the edges do not connect platform and depot events at the same station, thus eliminating the possibility of coupling a unit without taking a subtrip. Again, we assume a minimum duration on decoupling edges. This is modelled by connecting these edges to the target depot-station after the decoupling is completed. Note that the introduction of decoupling edges implicitly duplicates the number of subtrip edges that terminate at depot-stations. All coupling and decoupling edges are assigned an additional cost c_4 to reflect the price one must pay to perform a coupling. By counting the number of coupling edges used in any path from the source to the sink, one can determine how many times the unit is coupled and/or decoupled from a trip sequence. All (de)coupling edges must be adjusted by the dual contribution from Constraints (9) since any such edge indicates the consumption (or release) of depot track capacity. This dual contribution on any decoupling edge is given by $-l_u \hat{\nu}_a$, for subtrip $a \in \mathcal{A}$ where $\hat{\nu}_a = \sum_{a' \in \mathcal{A}_d^+} \nu_{a'}$. Here \mathcal{A}_d^+ gives the set of all subtrips in \mathcal{A} arriving at depot $d = \text{depot}(a) \in \mathcal{D}$ whose arrival time is at least as large as that of subtrip $a \in \mathcal{A}$. Similarly, the depot track dual contribution on any coupling edge is $l_u \hat{\nu}_a$, where $\hat{\nu}_a = \sum_{a' \in \mathcal{A}_d^-} \nu_{a'}$. The set \mathcal{A}_d^- is the set of all subtrips departing from $d \in \mathcal{D}$ a departure time at least the arrival time of $a \in \mathcal{A}$.

Constructing such a network allows units to perform non-consecutive subtrips, include the

coupling costs, and respect the minimum coupling durations. The last feature allows us to monitor when a unit type will be available to perform certain subtrips if (de)coupling decisions have been made. The shortest path in this network identifies the column with minimum reduced cost. Figure 7 shows how the network of Figure 6 can be extended to include depot subtrips as well as (de)coupling edges. Stations with depot tracks have depot event nodes and are coloured in light gray. Note that all (de)coupling edges have not been included; however, an example of each can be seen in two of the three example paths provided. The solid black path depicts a unit which is coupled to the second outgoing subtrip at depot $d_2 \in \mathcal{D}$. This unit performs four consecutive subtrips. The dashed path also illustrates a unit that performs four consecutive subtrips; however, this unit performs the first outgoing subtrip at station $d_2 \in \mathcal{D}$. This unit is decoupled after its fourth subtrip. In contrast, the gray path gives an example of a unit that stays at depot-station $d_1 \in \mathcal{D}$ during the planning period. As in Figure 6, a selection of arc weights have been included for expository purposes.

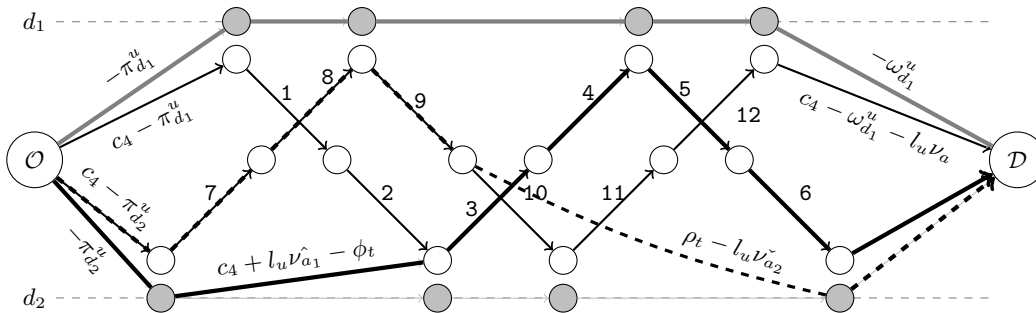


Figure 7: A simplified example of the acyclic time-space network showing some of the possible coupling arcs. Only two of the stations have an associated depot. Three possible unit paths have been highlighted.

In the absence of unit specific restrictions (e.g., a distance-to-maintenance), the subproblem is a standard shortest path problem on an acyclic graph. The shortest path can therefore be found using a directed acyclic graph shortest path algorithm with linear time complexity. The nature of the time-space network in Figure 7 is such that we can also satisfy requests which ask specific units to end at specific depots. Deadheading trips can also be included. Including such a trip simply involves adding an edge between station depots. Note that the network may contain negative edge weights; however, negative cycles cannot exist due to its acyclic nature.

The main motivation for considering a path based formulation is, however, to allow the incorporation of unit specific requirements. In Section 6 we consider the situation in which units have distance-to-maintenance restrictions. When including such a requirement, generating path variables entails solving the more difficult Shortest Path Problem with Resource Constraints (SPPRC), see, e.g., [23, 17], as we must monitor the total length (in km) when generating a sequence of subtrips for a unit with a distance-to-maintenance restriction. A standard label setting algorithm is used to solve this SPPRC. This particular problem has just one resource, the distance travelled, and this is constrained to be less than the remaining distance-to-maintenance. A label setting algorithm relies on a resource extension function. When solving the subproblem for a specific unit, if adding a subtrip to a partial path results in violation of the resource constraint, the path is pruned. Label setting algorithms also rely heavily on dominance to remove suboptimal partial paths. When considering distance-to-maintenance, one label dominates another if the partial path associated with the first label has a cost which is at least as good as that of the second label, and has a remaining distance-to-maintenance which is at least that of the second label. We do not describe this algorithm in detail as the implementation is quite straightforward considering we only have one resource and the underlying graph is a directed acyclic graph. We refer the interested reader to [22] for details on solving the SPPRC. Note that, on any given day, there *may be* several units for which the distance-to-maintenance requirement is unlikely to be violated. For such units the subproblem is solved by unit type, while for maintenance critical units, the subproblem is

solved by individual unit.

5.3. Branching

The column generation process terminates with an optimal solution to the RRMP. If the variables satisfy the integrality constraints, a feasible solution to the RSRP has been found, and we can update the upper bound and prune the current branch. This, however, does not always happen, and we need to define branching rules in order to enforce integrality.

Branching on path variables in the RRMP is not preferable in general. Forcing a fractional path variable to take the value zero or one in separate branch nodes complicates the subproblem. In the zero branch the subproblem will (without alterations) likely return the forbidden path as a new variable. Thus, in order to disallow specific paths we need to solve the *kth-shortest (resource constrained) path problem*, just to make sure that we have the shortest path that is not forbidden by branching rules. In addition, the efficiency of branching on paths is uncertain as the total number of paths is extremely large. We propose branching on certain characteristics of the paths instead, which is easier to handle in the subproblems.

We propose two sets of branching rules sufficient to force integrality. The first enforces integrality on the number of paths originating and terminating at a depot. In an integer solution, the number of paths leaving a depot (as the first action) can never be fractional. For example, if the sum of units that depart from a station is fractional, say $f \in \mathbb{R}$, then we create two branches where the sum is $\leq \lfloor f \rfloor$ on the first branch and $\geq \lceil f \rceil$ in the other. The same holds for the number of paths entering (as the last action) a depot. This rule can be applied on two levels, either for a specific unit type or aggregated over all types. Our BAP algorithm imposes this rule on an aggregated level where possible, otherwise on a specific unit type.

Even if the paths respect the integrality requirements above, the solution may still be fractional. The second set of branching rules enforces integrality on subtrip coverage. The subtrip cover is equal to the sum of paths that service a subtrip. We therefore ensure that the total subtrip cover, say $f \in \mathbb{R}$, is integer by imposing that the cover is $\leq \lfloor f \rfloor$ in one branch and $\geq \lceil f \rceil$ in the other. This branching rule can also be applied on two different levels. Where possible, we aggregate over unit types, otherwise we create a branch based on a specific unit type.

In the absence of unit specific constraints, and strictly positive (de)coupling costs, the above two branching strategies collectively provide a complete branching strategy.

Theorem 1. *In the absence of unit specific constraints, and assuming strictly positive (de)coupling costs, the aforementioned branching rules provide complete branching strategy.*

Proof. Assume we have an optimal solution to Model (1)–(13). Furthermore, assume without loss of generality that no two paths are identical.

We prove, by contradiction, that every subtrip is covered by integral paths. If all subtrips are covered by integral paths, then it follows naturally that the paths flowing in and out of depot sources and target nodes are integral.

Assume that an optimal solution is found but one (or multiple) subtrip is covered by multiple fractional paths, say p_1 and p_2 . These paths must diverge before or after some subtrip, otherwise they are identical. Assume the paths diverge after some subtrip, the other case is analogous. The case is illustrated in Figure 8.

After visiting a subtrip s_1 paths can either continue on the successor subtrip s_2 or decouple into the associate depot. Thus one of them will decouple. However, if a fraction of the flow decouples, then a different fractional path p_3 must be coupled on the successor path in order to maintain the integral sum of s_2 . We note that the fractional path which is leaving s_1 or entering s_2 can consist of multiple paths, but the sum is fractional. We also note that, without loss of generality, paths p_1 , p_2 or p_3 can further be a group of paths.

We now argue that this cannot be an optimal solution since a better alternative exists. Consider replacing p_2 and p_3 with new paths p_4 and p_5 , where these paths are the results of swapping the subpaths of p_2 and p_3 after the (de)coupling position. Path p_4 now continues along p_1 on s_1 and s_2 while path p_3 simply stays in the depot instead of coupling on to s_2 . The new resulting

solution (p_1, p_4, p_5) omits a coupling and decoupling penalty, thus obtaining a lower cost. Note that swapping subpaths is always possible in the absence of unit specific requirements. Thus we arrive at a contradiction, as the solution was assumed to be optimal. \square

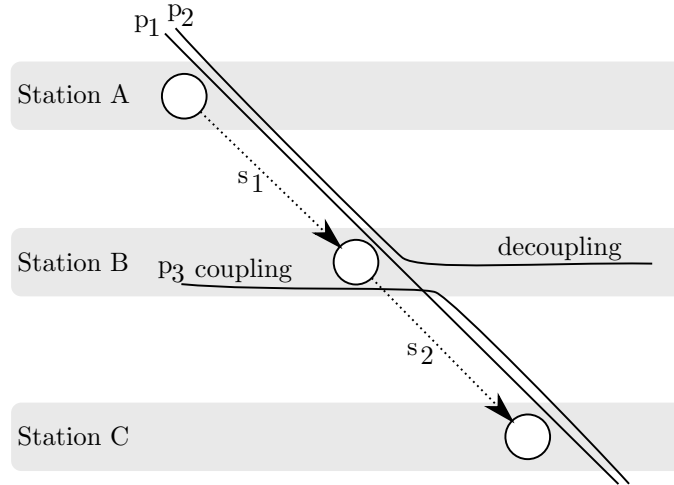


Figure 8: Illustration of a subset of a rolling stock schedule. Three fractional paths are shown covering two subtrips in the same trip sequence.

When unit specific constraints are considered, the above two branching strategies are insufficient. In particular, it may not necessarily be possible to swap subpaths of units. Swaps can only be made if they respect the individual requirements of the two units involved. As such, to ensure completeness of the branching procedure, in this case we define a third branching strategy based on constraint branching. This technique was first proposed in [31] for set partitioning problems. Instead of branching on original or subproblem variables, which often is imbalanced or ineffective, the authors show a more balanced approach is to branch on a characteristic of the problem, where any modifications can be easily incorporated in the variable generation process. For the set partitioning problem, the authors observe that, in an optimal integer solution, any two constraints are either covered by the same variable, or they are covered by different variables. Branching on this characteristic leads to a situation in which multiple variables are potentially banned in each branch, thus leading to a more balanced tree. For this particular problem we can observe that in any optimal solution to the RSRP, any subtrip is either covered by a unit or it is not. In the branching process we can therefore, if necessary, force particular units to perform particular subtrips. As the cover of a subtrip is allowed to be greater than one, however, this weakens the power of the constraint branch. Furthermore, applying the branching scheme on specific units introduces two main disadvantages. Firstly, such a rule will not have a significant effect since we have many similar train units. Secondly, variations of the subproblem must be solved for every unit depending on its individual branching constraints.

We prioritize the branching candidates by always favouring the first set over the second, and by preferring aggregation over unit specific rules. When considering unit specific requirements, we include the third branching strategy as a last resort. To reduce the size of the BAP tree, branching candidates are considered using a strong branching approach. In this work, we adopt a variant of so-called *Full Strong Branching*, where all possible branching candidates are generated in each branching node and ordered by fractional distance, see e.g., [2]. In our variant, not all candidates are considered; the candidates are evaluated in order until no better candidate has been found for a certain number of iterations.

5.4. Lower and Upper Bounds

To improve the performance of the algorithm we incorporate additional procedures to provide improved bounds on the solution quality. The first of these aims to quickly find an upper bound for the optimal solution and involves solving a model of the form (1)–(13), where the column set is restricted to those that have been generated while solving the LP relaxation, [to optimality](#). This procedure is run as soon as the LP relaxation has been solved to optimality and uses the commercial solver CPLEX. This much smaller integer program typically takes a few seconds to solve and in many cases provides a feasible integer solution to Model (1)–(13) of good quality. The second technique attempts to provide a tighter lower bound than that obtained the optimal solution to the LP relaxation, and involves solving the MIP formulation from [18]. This formulation is stated in full in Appendix A. Recall that the formulation of [18] only focuses at the unit type level and, via a flow based approach, assigns compositions to subtrips. The model can be solved independently of what we propose, and we solve it prior to running the BAP algorithm. It is important to note that this lower bound comes in the form of an integer solution and is found very quickly, in at most a few seconds. In the absence of unit specific requirements, this would also be the value of the optimal integer solution obtained using our path based approach. When unit specific requirements are included, however, the unit type flows obtained from the model in [18] may not necessarily decompose into a set of feasible individual unit paths. In such cases, it is therefore only a lower bound, but likely to be significantly tighter than that which is provided by the LP relaxation.

6. Computational Results

This section evaluates on the performance of the proposed BAP algorithm on real-life data provided by DSB S-tog and is divided into four subsections. First, Section 6.1, summarizes the base test instances that were provided by DSB S-tog and used in the experiments. The remaining three subsections highlight the method’s applicability to different problems. Section 6.2 addresses disruption management and presents the results of two disruption case studies. Section 6.3 summarizes the results of applying the methodology to the tactical level problem of rolling stock planning. Finally, Section 6.4 demonstrates the model’s ability to handle unit specific requirements by performing some experiments in which a distance-to-maintenance requirement must be observed.

6.1. Test Instances

The test instances are comprised of a set of timetables (and associated passenger demands) that were used by DSB S-tog during the spring of 2014. DSB S-tog adopts a weekly periodic timetable. All weekdays follow the same timetable, while Saturday and Sunday have a different one. On Friday and Saturday there are, however, additional night-trains. This results in four different timetables on which we can investigate our solution method. These are the Monday-Thursday timetable (Mon), the Friday timetable (Fri), the Saturday timetable (Sat), and the Sunday timetable (Sun). Table 5 summarises the key characteristics of the instances, while Figure 9 gives a detailed overview of DSB S-tog’s network and the lines operated.

Name	Stops	$ \mathcal{T} $	$ \mathcal{T}' $	Lines
Fri	28 719	1 086	3 259	A,B,Bx,C,E,F,H
Sat	20 474	706	1 795	A,B,C,F
Sun	19 919	690	1 753	A,B,C,F
Mon	28 017	1 074	3 196	A,B,Bx,C,E,F,H

Table 5: Four timetable and demand datasets provided by DSB S-tog. The columns respectively show the instance names, total number of stops, the subtrips, the number of non-aggregated train movements, and the lines that are operated.

Scenario #1 - Finger	
Contingency Plan	A.3.10 (Valby - Frederikssund)
Description	One track is blocked on one of the fingers
Cancelled lines	H
Turned lines	C1
Unchanged	A1, A2, B1, B2, Bx, C2, E1, E2

Table 6: Finger disruption: The disruption blocks one track between Valby and Frederikssund stations.

Scenario #2 - Central Corridor	
Contingency Plan	A.0.16 (København H - Østerport)
Description	One track is blocked in the central segment
Cancelled lines	A2, B2, Bx, C1, E2, H
Turned lines	A1, B1, C2, A1, C1
Unchanged	E1

Table 7: Central corridor disruption: The disruption blocks one track between København H and Østerport stations

As discussed in Section 2, a subtrip indicates the movement of a train service between two stations, where the delimiting stations permit composition changes to occur. A timetable can, however, specify movements between stations where the destination station does not allow this. There is no need to consider such a movement as an individual subtrip, since no composition change can occur. We therefore aggregate such a movement to its consecutive movement (and continue to do so until a destination station with a depot is reached). The resulting aggregated movement is then a subtrip using our terminology. Consequently, the set \mathcal{T} contains aggregated movements – or subtrips – as specified in the timetable. To indicate the level of aggregation, we report, using the set \mathcal{T}' , the number of individual, non-aggregated movements in the timetable (e.g., including those between stations where composition changes cannot occur). Table 5 indicates that this leads to a significant reduction in the number of subtrips that must be considered. We note that considering non-aggregated instances can be misleading as the complexity of the instance is related to the number of *real* shunting decisions that can be made. In Table 5 the number of stops refers to the number of times during the day a train stops at a platform to let passengers board or alight.

Recall that the essential differences between rolling stock planning and the RSRP are the timetable, the objective function, and the subtrip demands. For the disruption management case studies, we utilize the modified timetables operated during each of the disruptions; however, due to a lack of accurate data, we also use the forecast passenger demands during a disruption. The recovery problem is, however, much more constrained in terms of the available time in which a solution must be found.

6.2. Disruption Management

For the given timetables, the number of possible disruption scenarios is very large as there can be multiple points of failure, different disruption durations, different possible start times for a disruption, and different levels of severity. In order to analyse the performance of the proposed algorithm we find it sufficient to consider two different types of disruption, with different severities. The topology of the infrastructure is star-shaped (see Figure 9), which means that two interesting cases appear: A disruption in the central corridor (between Dybbølsbro and Østerport), or a disruption along one of the fingers. In the central corridor there is one track in both directions, and all lines through this corridor must share the track capacity. The timetables operated during a disruption are generated according to existing contingency plans. We test different start times and durations of both disruption types. Table 6 and 7 describe the two selected cases.



Figure 9: DSB S-tog network overview

<i>Inst</i>	Disruption	$ \mathcal{T} $	Time (s)	<i>Cols</i>	Quality				
					<i>Gap</i> ₁	<i>Gap</i> ₂	Cover	Seat	Mileage
Fri	9:00-10:00	823	5	2,563	0.9%	3.4%	100.0%	99.0%	122.9%
Fri	9:00-13:00	814	6	2,608	0.0%	3.2%	100.0%	99.4%	124.8%
Fri	11:00-12:00	703	5	2,234	0.3%	3.8%	100.0%	99.1%	121.3%
Fri	11:00-15:00	692	5	2,483	0.3%	2.1%	100.0%	99.0%	122.7%
Fri	15:00-16:00	456	3	1,456	0.0%	0.8%	100.0%	98.9%	118.3%
Fri	15:00-19:00	442	3	1,362	0.0%	1.2%	100.0%	99.2%	118.8%
Mon	9:00-10:00	807	5	2,642	0.1%	2.6%	100.0%	98.9%	126.1%
Mon	9:00-13:00	798	6	3,391	0.6%	2.8%	100.0%	99.1%	126.3%
Mon	11:00-12:00	687	6	2,460	0.9%	4.0%	100.0%	98.8%	124.5%
Mon	11:00-15:00	676	5	2,854	0.5%	2.9%	100.0%	98.9%	121.6%
Mon	15:00-16:00	440	5	1,626	0.0%	2.4%	100.0%	98.2%	119.0%
Mon	15:00-19:00	426	3	1,416	0.7%	2.1%	100.0%	98.2%	121.1%

Table 8: The results obtained solving the first type of disruption case. The first columns show the instance, disruption duration, number of subtrips and runtime. *Cols* shows the number of generated columns. Columns *Gap*₁ and *Gap*₂ show the optimality gaps to the lower bound from [18] and the LP relaxation of Model (1)–(13), respectively. Finally, the last few columns measure the quality of the solution in terms of covered subtrips, covered seats and train unit mileage.

In our experiments, we consider disruption durations of between one and four hours with three different starting points: 09:00, 11:00, and 15:00. The selected periods overlap with the peak periods in different ways. Instances Sat and Sun are excluded from consideration as they are arguably easier to solve. Given the type of disruption (finger, central corridor), duration and starting point, an updated timetable to operate during the disruption is obtained using the described contingency plans. The planned rolling stock schedule is assumed to have been used without modification up until the starting point of the disruption, and the remainder of the day must be solved. We make the assumption that the duration of the disruption is known in advance. Disruptions with an unknown duration can be handled using a rolling horizon scheme, whereby an updated timetable is obtained and an RSRP is solved whenever new information becomes available. No real-life data could be obtained for doing experiments with maintenance restrictions. However, in our final experiments we generate a number of realistic datasets for benchmarking this extension.

All tests have been performed on Intel(R) Xeon(R) CPU X5550 @ 2.67GHz with 24GB ram running Ubuntu Linux 14.04. After observing steadier convergence with the COIN-OR solver (CLP) compared to CPLEX, it was decided to use this as the LP solver for the RRMP.

The results of the experiments are shown in Tables 8 and 9. The algorithm was set to terminate and report the best found solution within five minutes or when an optimality gap of 1% is proven. Since we solve the remainder of the day this means that longer durations and later starting points result in smaller instances. Note that (de)couplings are given a high penalty during disruptions, thus favouring fewer composition changes over seat and mileage cover. As an indication, the cost of a (de)coupling is set to 1,000. Cancelling a subtrip incurs a cost of 50,000. One missing seat is penalized by 0.1, and this is also the cost incurred in driving a unit one kilometre. Finally, an imbalance in a depot at the end of the day is penalized by 10,000 for every missing unit.

In Table 8 the results of activating the relevant contingency plan *A.3.10* with multiple duration and starting points can be seen. We obtain very good solutions within a few seconds. Due to a decrease in the number of subtrips, a low runtime is expected. Again, the lower bound of [18] is noticeably tighter than that obtained using the LP relaxation. In all cases the final solutions are found after using the upper bound procedure described in Section 5.4; i.e., after solving a MIP using the columns generated during the convergence of the optimal solution to the LP relaxation. No branching is needed in the BAP algorithm. A full cover is obtained in all cases; this means that all trips in the disrupted timetable can be covered. The cover metric does not include the trips cancelled due to switching to the contingency plan.

Table 9 shows the results of activating the contingency plan *A.0.16* with multiple duration

<i>Inst</i>	Disruption	$ \mathcal{T} $	Time (s)	<i>Cols</i>	Quality				
					<i>Gap</i> ₁	<i>Gap</i> ₂	Cover	Seat	Mileage
Fri	9:00-10:00	836	11	4,096	0.3%	15.1%	99.8%	98.9%	122.4%
Fri	9:00-13:00	782	18	6,068	0.5%	14.7%	99.8%	98.7%	119.4%
Fri	11:00-12:00	713	8	3,051	0.6%	16.7%	99.8%	98.9%	122.1%
Fri	11:00-15:00	657	10	3,939	0.2%	14.8%	99.7%	98.4%	117.5%
Fri	15:00-16:00	468	6	2,379	0.3%	2.1%	99.6%	97.2%	113.8%
Fri	15:00-19:00	402	5	1,727	0.0%	1.9%	99.8%	98.4%	115.3%
Mon	9:00-10:00	820	13	4,129	0.4%	2.7%	99.8%	98.4%	121.6%
Mon	9:00-13:00	766	14	5,827	0.1%	2.5%	99.8%	98.8%	118.3%
Mon	11:00-12:00	697	10	3,665	0.5%	16.5%	99.8%	98.8%	122.4%
Mon	11:00-15:00	641	8	4,066	0.2%	14.3%	99.7%	98.4%	118.6%
Mon	15:00-16:00	451	6	2,239	0.0%	1.5%	99.6%	97.5%	119.1%
Mon	15:00-19:00	385	6	2,053	0.2%	1.9%	99.8%	97.2%	118.2%

Table 9: Results of solving the second type of disruption case. The first columns show the instance, disruption duration, no of subtrips and runtime. *Cols* shows the number of generated columns. Columns *Gap*₁ and *Gap*₂ show the optimality gaps to the lower bound from [18] and the LP relaxation of Model (1)–(13), respectively. Finally, the last few columns measure the quality of the solution in terms of covered subtrips, covered seats and train unit mileage.

and starting points. Good solutions to these problems are also found within a few seconds, and no branching is necessary. The upper bound procedure from Section 5.4 always provides solutions within the 1% optimality threshold. Compared to Table 8, the lower bound of [18] is much tighter. In the worst case the improvement is approximately 15% compared to the objective value of the optimal solution to the LP relaxation. The larger gaps to the bound provided by the LP relaxation are probably correlated with the reduced trip cover. We observe that some trips are now cancelled in the disrupted timetable, and a fractional, optimal solution to the LP relaxation is able to cover more trips and seats using fewer couplings and mileage. The mileage usage is still comparable to our previous results. The slightly lower seat cover is mainly due to the fact that additional trains need to be cancelled.

6.3. Tactical Level Planning

We now demonstrate that Model (1)–(13) can also be applied to the tactical level planning problem of rolling stock scheduling and that the proposed BAP algorithm can be extended to include unit specific requirements. From a tactical planning perspective, the most important objective is the number of subtrip cancellations. All other objectives are of small importance in comparison. We only expect to see cancellations during a disruption. After cancellations, we consider end-of-day balance deviations to be the second most important as deadheading units is an expensive and undesirable activity. Finally, a good balance between demand cover, operational cost and the number of (de)couplings is sought. In the first experiment we therefore analyse the performance of the algorithm solely considering the number of cancellations. One cancellation incurs a cost of 100,000. An insignificantly small penalty of value one is, however, incurred per (de)coupling since preliminary experiments show that the solution method performs worse without this. Without a coupling cost the number of alternative optimal paths in the subproblem increases drastically; the number of different paths (with the same cost) negatively affects the integrality of the master problem. The results for this test are shown in Table 10 for the four timetables. The results show that we are able to solve the planning instances to optimality within a few seconds. In all cases there is a gap of 0% between the optimal integer solution and the bound provided by the LP relaxation. This suggests that our formulation has some good properties using the considered objectives. All instances were solved to optimality before branching could occur. This experiment provides evidence that the model performs well when the sole goal is to find a certain (integral) cover, i.e., when all subtrip compositions are given beforehand.

<i>Inst</i>	Horizon	$ \mathcal{T} $	Time (s)	<i>Cols</i>	<i>Gap</i>
Fri	4:40-27:10	1,086	6	1,246	0.0%
Sat	2:00-27:10	706	1	411	0.0%
Sun	2:00-25:38	690	1	350	0.0%
Mon	4:37-25:38	1,074	7	1,584	0.0%

Table 10: The results obtained considering only cancellation costs and a small cost (of 1) for (de)couplings. The horizon indicates the planning horizon of the instance (given in hours after midnight). *Gap* indicates the optimality gap to the LP relaxation of Model (1)–(13).

<i>Inst</i>	\mathcal{T}	Time (s)	<i>Cols</i>	<i>Gap</i> ₁	<i>Gap</i> ₂	Cover
Fri	1,086	31	3,328	4.0%	13.6%	100.0%
Sat	706	1	514	0.0%	0.0%	100.0%
Sun	690	2	472	0.0%	2.8%	100.0%
Mon	1,074	19	2,678	1.5%	9.1%	100.0%

Table 11: The results obtained only considering cancellation costs, a small shunting cost and a seat shortage cost. The elapsed time is measured in seconds. *Gap*₁ and *Gap*₂ show optimality gaps to the lower bound method from [18] and the LP relaxation of Model (1)–(13), respectively. *Cover* shows the percentage of covered seats in the found solution.

A minimal seat-shortage is desirable and experiments performed using a small (de)coupling cost (value of 100) and different shortage costs are shown in Table 11. The results show that the algorithm has difficulty solving the problem to optimality. Seat cover is at 100% which means seat shortage contributes with a zero cost to the objective. A 5% termination gap was used since the algorithm has difficulty obtaining solutions close to optimality (i.e. a few percent), let alone proving optimality, for all datasets. This is even the case if we allow a time limit of one hour. The gap is therefore solely caused by the number of (de)couplings performed. The relatively large gap to the LP relaxation is not unexpected since fractional unit paths are able to cover seat demand more cost-efficiently than integral paths, which have to pay full (de)coupling penalties. No branching was performed. In a planning context, the number of (de)couplings is not a primary concern compared to scheduling new (de)couplings during the day of operation.

In planning the goal is to meet the expected demand. A seat-coverage of 100% is ideal; however, if the cost of covering 99% is significantly less, then this is also an acceptable solution. Table 12 shows the seat cover using varying seat shortage costs. As expected, the seat cover increases as the seat-shortage penalty increases.

Minimizing cancellations, shunting operations and seat shortages can lead to expensive schedules since train services can be executed with too much capacity. Minimizing the mileage is therefore also essential in order to avoid unnecessary wear and tear. In the final planning benchmark we use varying mileage costs. The results are shown in Table 13. Note, that *mileage cover* is relative to the travel-distance of assigning one unit to all subtrips (including dead-heading). Thus 100% is the minimum if no cancellations are made. Using different penalties the total mileage decreases as the mileage penalty increases. As we aim for a high seat cover the trade-off is between the number of (de)couplings and mileage. The total runtime ranges up to a few minutes, and with

Instance/Penalty	0.0	0.05	0.1	1.0
Fri	87.04%	99.94%	99.95%	100.00%
Sat	96.90%	100.00%	100.00%	100.00%
Sun	98.16%	100.00%	100.00%	100.00%
Mon	83.14%	99.95%	100.00%	100.00%

Table 12: Seat demand coverage obtained using increasing seat shortage penalties.

<i>Inst</i>	<i>Pen.</i>	Time (s)	<i>Cols</i>	<i>Gap</i> ₁	<i>Gap</i> ₂	Seat	Mileage	(De)couplings
Fri	0.01	100	6,068	2.9%	12.4%	100.0%	126.5%	282
Fri	0.10	113	6,286	1.5%	10.1%	100.0%	126.2%	278
Fri	1.00	590	14,158	0.9%	6.9%	100.0%	119.3%	294
Sat	0.01	5	1,295	0.0%	0.0%	100.0%	100.6%	116
Sat	0.10	5	1,317	0.0%	0.0%	100.0%	100.6%	116
Sat	1.00	20	3,860	0.0%	0.0%	100.0%	100.6%	116
Sun	0.01	4	1,180	0.0%	2.8%	100.0%	100.8%	120
Sun	0.10	4	1,316	0.0%	2.4%	100.0%	100.8%	120
Sun	1.00	23	3,842	0.0%	1.2%	100.0%	100.8%	120
Mon	0.01	45	4,850	1.4%	9.1%	100.0%	126.0%	274
Mon	0.10	48	5,126	0.7%	8.2%	100.0%	125.8%	272
Mon	1.00	255	10,695	0.7%	7.3%	100.0%	119.1%	296

Table 13: The results obtained using increasing mileage cost penalties. A termination gap of 5% was set. The columns show the instance, mileage penalty, runtime, number of columns generated, optimality gap to the lower bound of [18], the optimality gap to the LP relaxation of Model (1)–(13), seat coverage, mileage cover and number of (de)couplings.

a 5% optimality tolerance no branching was performed. The optimality gap to the LP relaxation also improves as the (de)coupling penalty diminishes (relative to mileage cost), and more columns are necessary to prove LP optimality. This is because mileage and (de)coupling become more equally priced in the objective. To provide an indication of the branching performance of the algorithm, Table 14 summarizes the same tests, where the optimality tolerance is lowered to 0.1% and a one hour time limit is enforced. The results show that half of the instances time out and that there is not a dramatic improvement in the optimality gap; no new solutions are found compared to Table 13. The branching strategies appear to struggle to make an impact on the lower bound. Similar observations can be made when lowering the optimality tolerance for the larger disruption cases. However, for the smaller ones, branching is typically not as critical, and therefore these can be solved to optimality.

The planned cover and mileage at DSB S-tog is consistent with our results and only differ slightly in value. Table 15 shows the planned and realized schedules for the instances we consider. Note that the realized schedule summarizes actually what happened in practice when using the corresponding planned schedule. DSB S-tog has optimization tools available for planning (but not rescheduling), and the planned schedule has been obtained using such tools. Thus, we cannot expect to find plans that are significantly better. In general, the planners have achieved a high seat cover and also a relatively low mileage cover at the cost of more (de)couplings.

6.4. Unit Specific Requirements

As a last experiment we show that the proposed algorithm can naturally enforce unit specific requirements, regardless of the planning level. To demonstrate this we give some results of adding mileage restrictions in the form of a distance-to-maintenance to certain units. Recall that this requirement prevents a unit from being assigned a sequence of subtrips if the combined distance of the subtrips exceeds the remaining distance-to-maintenance level. To give some context, we consider the sanding inspection requirement on the units in DSB S-tog’s fleet. This inspection is crucial as it ensures proper function of the wheel slide protection system. All units must undergo this inspection every ten megametre.

We were unable to obtain real-life data and instead generate several realistic instances. A single unit usually drives up to one megametre during a normal weekday and, on average, half a megametre. Ideally, at the beginning of the planning horizon every unit is assigned a random distance-to-maintenance having a value between zero and ten megametres; however, we only need to assign the random distance-to-maintenance on a selected subset of the fleet using an appropriate

<i>Inst</i>	<i>Pen.</i>	Time (s)	<i>Cols</i>	<i>Nodes</i>	<i>Gap</i> ₁	<i>Gap</i> ₂	Seat	Mileage	(De)couplings
Fri	0.01	3,600	42,392	837	2.9%	12.0%	100.0%	126.5%	282
Fri	0.10	3,600	38,589	684	1.5%	10.1%	100.0%	126.2%	278
Fri	1.00	3,600	32,830	144	0.9%	6.5%	100.0%	119.3%	294
Sat	0.01	5	1,295	1	0.0%	0.0%	100.0%	100.6%	116
Sat	0.10	5	1,317	1	0.0%	0.0%	100.0%	100.6%	116
Sat	1.00	20	3,860	1	0.0%	0.0%	100.0%	100.6%	116
Sun	0.01	5	1,180	2	0.0%	2.8%	100.0%	100.8%	120
Sun	0.10	7	1,316	2	0.0%	2.4%	100.0%	100.8%	120
Sun	1.00	34	3,842	2	0.0%	1.2%	100.0%	100.8%	120
Mon	0.01	3,600	62,980	554	1.4%	9.0%	100.0%	126.0%	274
Mon	0.10	3,600	64,489	515	0.7%	8.0%	100.0%	125.8%	272
Mon	1.00	3,600	54,361	144	0.7%	7.0%	100.0%	119.1%	296

Table 14: The results obtained using increasing mileage cost penalties. A termination gap of 0.1% and a time limit of 3,600s were set. The columns show the instance, mileage penalty, runtime, number of BAP nodes, number of columns generated, optimality gap to the lower bound of [18], the optimality gap to the LP relaxation of Model (1)–(13), seat coverage, mileage cover and number of (de)couplings.

<i>Inst</i>	Planned		Realized		
	Unit Cover	Demand cover	Unit Cover	Demand cover	Couplings
Fri	117.44%	99.91%	117.75%	99.32%	371
Sat	112.41%	99.96%	112.41%	99.81%	205
Sun	109.54%	99.88%	109.71%	99.69%	186
Mon	116.08%	99.99%	116.09%	91.71%	459

Table 15: Shows the statistics of the planned and realized solution for the datasets. The number of couplings is unavailable in the planned solution and is therefore estimated based on the paths in the realized solution.

Probability	Time (s)	<i>Cols</i>	<i>Gap</i>	(De)couplings	Seat	Mileage
5%	464	15,135	1.4%	304	100.0%	118.0%
5%	461	15,135	1.4%	304	100.0%	118.0%
5%	450	17,227	2.0%	308	100.0%	118.1%
5%	351	14,441	1.5%	304	100.0%	118.3%
5%	273	11,832	1.9%	306	100.0%	117.9%
10%	401	18,982	1.6%	298	100.0%	120.3%
10%	400	18,982	1.6%	298	100.0%	120.3%
10%	413	16,196	2.6%	312	100.0%	118.2%
10%	621	20,579	1.6%	302	100.0%	119.1%
10%	526	20,358	2.1%	310	100.0%	117.7%
15%	690	25,389	1.5%	302	100.0%	118.8%
15%	686	25,389	1.5%	302	100.0%	118.8%
15%	490	18,522	2.0%	308	100.0%	118.1%
15%	346	16,392	2.6%	310	100.0%	118.7%
15%	222	14,451	2.2%	308	100.0%	118.5%

Table 16: Results of maintenance cases. The first column shows the probability of a unit receiving a limited mileage during the planning horizon. The other columns respectively show runtime, number of generated columns, optimality gap to the LP relaxation of Model (1)–(13), number of (de)couplings, seat cover and mileage cover.

random probability (5-15%). As mentioned earlier, there is no reason to place restrictions on units that will never reach the limit. Hence, such units are, in our opinion, fully unrestricted in mileage and interchangeable. Using different random seeds we generate and solve different maintenance instances. Each maintenance instance is based on the Mon timetable, assuming a tactical level planning problem, and a mileage penalty of 1.0.

The results of the benchmark are shown in Figure 16. No branching was performed. Note that it is not possible to provide an exact comparison (to the results in Table 13) as we only find a heuristic solution with a proven low gap. With a noticeable increase in runtime, the demands of all instances were covered fully, and the total mileage is comparable to the original instance. The number of (de)couplings is similar, if not slightly higher. In addition some variation comes with the random selection of units to which a distance-to-maintenance threshold is set.

The optimal solution to the LP relaxation can be used as an indication of how much more constrained the fleet is. Here, compared to the original instance (Mon), we did not notice a significant increase in the relaxed objective value (i.e., overall cost) with the more restricted units. We conclude that the fleet is able to absorb the added restriction without incurring high penalties. The main aim of this maintenance extension is to highlight that the model can incorporate such individual requirements. As such, how the distance-to-maintenance is determined and to which units it is assigned is not deemed significantly important here.

7. Conclusions

In this paper we have proposed a Branch-and-Price algorithm for optimizing the Rolling Stock Rescheduling Problem. This has been tested on two real-life case studies provided by DSB S-tog. We have demonstrated that this model is not only applicable in a disruption management setting, but can also be applied on a tactical level when planning the rolling stock. Finally, we have also shown that the model and solution approach can incorporate individual unit requirements such as a distance-to-maintenance constraint.

The BAP algorithm is able to solve all instances to optimality, but this proves to be too time-consuming in certain settings. Computational experiments revealed that we can find solutions with a small proven optimality gap for the hardest instances within 20 seconds for the disruption cases and within a few minutes for the planning instances. In general, [the tests show that](#) the algorithm is able to find good solutions very fast, but on the other hand it is not able to close the

optimality gap efficiently. An adopted MIP based lower bound efficiently improved the optimality gap. Further testing also revealed that the individual maintenance restrictions were well handled.

Several promising areas for future research exist. One obvious extension is to impose train composition and shunting restrictions. This would allow the train operators to more explicitly control how already running train compositions can be upgraded or reduced in order to improve robustness or feasibility. The proposed mathematical model mainly consists of several soft constraints with non-integral right hand sides. Enforcing hard constraints would limit the solution space and replacing the demand constraints with the most desired train composition would arguably improve the solution method. Finally, it would be interesting to extend the current models in order to model already used maneuvers at DSB S-tog during disruptions. Examples include allowing trains to turn earlier than planned, and allowing two trains to swap services when meeting at the same station.

Acknowledgements

The authors would like to thank the anonymous referees for their valuable comments and the questions that they raised. This feedback has been used to improve the quality of the paper.

Appendix A. Lower Bound Formulation

The MIP formulation that is used to obtain a tighter lower bound for the proposed path based formulation is the flow based model of [18]. This formulation assigns compositions to each of the subtrips while preserving the flow of train units. It does not determine individual itineraries for each of the train units; however, in the absence of unit specific requirements, any solution to this flow formulation can be decomposed into a set of train unit paths. For reference, we include all details of this formulation. To do this, additional variables and notation are needed. These are included in Tables A.17, Table A.18, and Table A.19. A binary variable q_t^c is introduced for every subtrip and composition combination. The variable takes on a value of 1 if composition c is chosen for subtrip t . In addition, the binary variable $r_{c,c'}^a$ tracks composition changes on connections. A connection is here two subtrips that are performed in sequence (in a trip sequence). The mathematical formulation is listed below.

q_t^c	Binary variable which takes value 1 iff composition c is chosen for covering trip t
$r_{c,c'}^a$	Binary variable which takes value 1 iff a transition from composition c to c' is chosen on connection a

Table A.17: Additional variables for the Lower Bound MIP model.

$c \in \mathcal{C}$	Set of all possible compositions, where a composition c defines a set of train units
$t \in \mathcal{T}_{\leftrightarrow}$	Set of all connections. A connection consists of two subtrips t_1, t_2 that are in sequence. The destination station of t_1 is naturally the origin station of t_2 .

Table A.18: Additional sets for the Lower Bound MIP model.

$in(a)$	First subtrip (incoming) in connection a
$out(a)$	Second subtrip (outgoing) in connection a
$source(t)$	Origin station of subtrip t
$target(t)$	Destination station of subtrip t
$dep(t)$	Departure time from origin of subtrip t
$arr(t)$	Arrival time at destination station of subtrip t

Table A.19: Additional shorthand notation for the Lower Bound MIP model.

$$\text{Minimize: } F(\mathbf{y}, \mathbf{q}, \mathbf{w}, \mathbf{r}) \quad (\text{A.1})$$

$$\sum_{c \in \mathcal{C}} \mathbf{q}_t^c = 1 - \mathbf{y}_t \quad \forall t \in \mathcal{T} \quad (\text{A.2})$$

$$\mathbf{q}_c^{in(a)} = \sum_{c' \in \mathcal{C}} \mathbf{r}_{c,c'}^a \quad \forall a \in \mathcal{T}_{\leftrightarrow}, c \in \mathcal{C} \quad (\text{A.3})$$

$$\mathbf{q}_c^{out(a)} = \sum_{c' \in \mathcal{C}} \mathbf{r}_{c',c}^a \quad \forall a \in \mathcal{T}_{\leftrightarrow}, c \in \mathcal{C} \quad (\text{A.4})$$

$$0 \leq inventory_d^u + \sum_{c \in \mathcal{C}} units_u^c \cdot \left(\sum_{\substack{t' \in \mathcal{T} \\ target(t')=d \\ arr(t') \leq dep(t)}} \mathbf{q}_{t'}^c - \sum_{\substack{t' \in \mathcal{T} \\ source(t')=d \\ dep(t') \leq dep(t)}} \mathbf{q}_{t'}^c \right) \quad \forall t \in \mathcal{T}, u \in \mathcal{U}, d = source(t) \quad (\text{A.5})$$

$$eod_d^u - \mathbf{w}_d^u \leq inventory_d^u + \sum_{c \in \mathcal{C}} units_u^c \cdot \left(\sum_{\substack{t \in \mathcal{T} \\ target(t)=d}} \mathbf{q}_t^c - \sum_{\substack{t \in \mathcal{T} \\ source(t)=d}} \mathbf{q}_t^c \right) \quad \forall u \in \mathcal{U}, d \in \mathcal{D} \quad (\text{A.6})$$

$$track_d \geq \sum_{u \in \mathcal{U}} l_u \cdot \left(inventory_d^u + \sum_{c \in \mathcal{C}} units_u^c \cdot \left(\sum_{\substack{t' \in \mathcal{T} \\ target(t')=d \\ arr(t') \leq arr(t)}} \mathbf{q}_{t'}^c - \sum_{\substack{t' \in \mathcal{T} \\ source(t')=d \\ dep(t') \leq arr(t)}} \mathbf{q}_{t'}^c \right) \right) \quad \forall t \in \mathcal{T}, d = target(t) \quad (\text{A.7})$$

$$\mathbf{q}_t^c \in \{0, 1\} \quad \forall t \in \mathcal{T}, c \in \mathcal{C} \quad (\text{A.8})$$

$$\mathbf{r}_{c,c'}^a \in \{0, 1\} \quad \forall a \in \mathcal{T}_{\leftrightarrow}, c, c' \in \mathcal{C} \quad (\text{A.9})$$

$$\mathbf{y}_t \in \{0, 1\} \quad \forall t \in \mathcal{T} \quad (\text{A.10})$$

$$\mathbf{w}_d^u \in \mathbb{Z}_0^+ \quad \forall u \in \mathcal{U}, d \in \mathcal{D} \quad (\text{A.11})$$

Constraints (A.2) ensure that exactly one composition is assigned to each subtrip, unless there is a cancellation. Some compositions may be invalid on certain subtrips (e.g., due to small platform lengths), these variables are assumed to be fixed to zero. Constraints (A.3) and (A.4) link the composition variables with the transition variables. Constraints (A.5) ensure that a non-negative number of units are in the depot after each departure. For each unit type the right hand side sums the initial inventory and the units moving in and out of the depot respectively. Constraints (A.6)

ensure that sufficient units of every type are available at the end of the day; missing units are captured in the w_d^u variables, which in turn are penalized in the objective function. Constraints (A.7) ensure that the capacity of every depot is respected at all times; i.e., at every subtrip arrival. Finally, Constraints (A.8)–(A.11) state the domains of the decision variables. The objective (A.1) is a function of the decision variables, reflecting the same objective as our previous formulation. The cancellation cost depends on the number of cancellations, y . The mileage and seatshortage costs depend on the assigned compositions, q . The end of day deviations, w , are penalized for every depot. Any (de)couplings are penalized using the r and q variables. The former are used to penalize whenever compositions change occur, while the latter are also needed for the first and last subtrips of a trip sequence.

References

- [1] Erwin Abbink, Bianca van den Berg, Leo Kroon, and Marc Salomon. Allocation of railway rolling stock for passenger trains. *Transportation Science*, 38:33–41, 2004.
- [2] Tobias Achterberg, Thorsten Koch, and Alexander Martin. Branching rules revisited. *Operations Research Letters*, 33(1):42–54, 2005. ISSN 01676377.
- [3] Arianna Alfieri, Rutger Groot, Leo Kroon, and Alexander Schrijver. Efficient circulation of railway rolling stock. *Transportation Science*, 40(3):378–391, 2006. ISSN 15265447.
- [4] Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh, and Pamela H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.
- [5] Nejib Ben-Khedher, Josephine Kintanar, Cecile Queille, and William Stripling. Schedule optimization at sncf: from conception to day of departure. *Interfaces*, 28(1):6–23, 1998. ISSN 1526551x.
- [6] Ralf Borndörfer, Markus Reuther, Thomas Schlechte, and Steffen Weider. A Hypergraph Model for Railway Vehicle Rotation Planning. In Alberto Caprara and Spyros Kontogiannis, editors, *11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 20 of *OpenAccess Series in Informatics (OASICs)*, pages 146–155, Dagstuhl, Germany, 2011. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-939897-33-0. doi: <http://dx.doi.org/10.4230/OASICs.ATMOS.2011.146>. URL <http://drops.dagstuhl.de/opus/volltexte/2011/3274>.
- [7] Gabriella Budai, Gábor Maróti, Rommert Dekker, Dennis Huisman, and Leo Kroon. Rescheduling in passenger railways: the rolling stock rebalancing problem. *Journal of Scheduling*, 13:281–297, 2010. ISSN 1094-6136. URL <http://dx.doi.org/10.1007/s10951-009-0133-9>.
- [8] Valentina Cacchiani, Alberto Caprara, and Paolo Toth. Solving a real-world train-unit assignment problem. *Math. Program.*, 124(1-2):207–231, 2010. ISSN 14364646.
- [9] Valentina Cacchiani, Alberto Caprara, Laura Galli, Paolo Toth, Leo Kroon, Gábor Maróti, and Leo Kroon. Railway rolling stock planning: Robustness against large disruptions. *Transportation Science*, 46(2):217–232, 2012. ISSN 15265447.
- [10] Valentina Cacchiani, Dennis Huisman, Martin Kidd, Leo Kroon, Paolo Toth, Lucas Veenlenturf, and Joris Wagenaar. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, 63:15–37, 2014.
- [11] Luis Cadarso and Ángel Marín. Robust rolling stock in rapid transit networks. *Computers & Operations Research*, 38(8):1131–1142, 2011. ISSN 1873765x.

- [12] Luis Cadarso, Ángel Marín, and Gábor Maróti. Recovery of disruptions in rapid transit networks. *Transportation research. Part E, Logistics and transportation review*, 53:15–33, 2013. ISSN 18785794.
- [13] Alberto Caprara, Leo Kroon, Michele Monaci, Marc Peeters, and Paolo Toth. Passenger railway optimization. In Cynthia Barnhart and Gilbert Laporte, editors, *Handbooks in Operations Research and Management Science*, volume 14, chapter 3, pages 129–187. Elsevier, 2007.
- [14] Jean-François Cordeau, François Soumis, and Jacques Desrosiers. A benders decomposition approach for the locomotive and car assignment problem. *Transportation Science*, 34(2):133–149, May 2000. ISSN 1526-5447. doi: 10.1287/trsc.34.2.133.12308. URL <http://dx.doi.org/10.1287/trsc.34.2.133.12308>.
- [15] Jean-François Cordeau, Guy Desaulniers, Norbert Lingaya, François Soumis, and Jacques Desrosiers. Simultaneous locomotive and car assignment at {VIA} rail canada. *Transportation Research Part B: Methodological*, 35(8):767 – 787, 2001. ISSN 0191-2615. doi: [http://dx.doi.org/10.1016/S0191-2615\(00\)00022-9](http://dx.doi.org/10.1016/S0191-2615(00)00022-9). URL <http://www.sciencedirect.com/science/article/pii/S0191261500000229>.
- [16] Jean-François Cordeau, François Soumis, and Jacques Desrosiers. Simultaneous assignment of locomotives and cars to passenger trains. *Oper. Res.*, 49(4):531–548, July 2001. ISSN 0030-364X. doi: 10.1287/opre.49.4.531.11226. URL <http://dx.doi.org/10.1287/opre.49.4.531.11226>.
- [17] Jacques Desrosiers, Yvan Dumas, Marius M. Solomon, and François Soumis. Chapter 2 time constrained routing and scheduling. In C.L. Monma M.O. Ball, T.L. Magnanti and G.L. Nemhauser, editors, *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, pages 35 – 139. Elsevier, 1995. doi: [http://dx.doi.org/10.1016/S0927-0507\(05\)80106-9](http://dx.doi.org/10.1016/S0927-0507(05)80106-9). URL <http://www.sciencedirect.com/science/article/pii/S0927050705801069>.
- [18] Pieter-Jan Fioole, Leo Kroon, Gábor Maróti, and Alexander Schrijver. A rolling stock circulation model for combining and splitting of passenger trains. *European Journal of Operational Research*, 174(2):1281–1297, 2006. ISSN 03772217.
- [19] Julie Groth, Daniel Potthoff, Jens Clausen, Dennis Huisman, Leo Kroon, Gábor Maróti, and Morten Nielsen. Disruption management in passenger railway transportation. In Ravindra Ahuja, Rolf Möhring, and Christos Zaroliagis, editors, *Robust and Online Large-Scale Optimization*, volume 5868 of *Lecture Notes in Computer Science*, pages 399–421. Springer Berlin / Heidelberg, 2009. ISBN 978-3-642-05464-8. URL http://dx.doi.org/10.1007/978-3-642-05465-5_18.
- [20] Julie Jespersen Groth and Jens Clausen. Optimal reinsertion of cancelled train lines. Technical report, 2006. Technical University of Denmark.
- [21] Julie Jespersen Groth, Daniel Potthoff, Jens Clausen, Dennis Huisman, Leo Kroon, Gábor Maróti, and Morten Nyhave Nielsen. Disruption management in passenger railway transportation. *Robust and Online Large-Scale Optimization*, pages 399–421, 2009.
- [22] Stefan Irnich. Resource extension functions: properties, inversion, and generalization to segments. *OR Spectrum*, 30(1):113–148, 2008. ISSN 0171-6468.
- [23] Stefan Irnich and Guy Desaulniers. Shortest path problems with resource constraints. In Guy Desaulniers, Jacques Desrosiers, and Marius M. Solomon, editors, *Column Generation*, pages 33–65. Springer US, 2005. ISBN 978-0-387-25485-2.

- [24] Leo Kroon and Dennis Huisman. Algorithmic support for railway disruption management. In Jo A.E.E. Nunen, Paul Huijbregts, and Piet Rietveld, editors, *Transitions Towards Sustainable Mobility*, pages 193–210. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-21191-1. doi: 10.1007/978-3-642-21192-8_11. URL http://dx.doi.org/10.1007/978-3-642-21192-8_11.
- [25] Zhiyuan Lin and Raymond S K Kwan. A branch-and-price approach for solving the train unit scheduling problem. *Transportation Research Part B: Methodological*, 94:97–120, 2016. ISSN 18792367, 01912615. doi: 10.1016/j.trb.2016.09.007.
- [26] Norbert Lingaya, Jean-François Cordeau, Guy Desaulniers, Jacques Desrosiers, and Francois Soumis. Operational car assignment at via rail canada. *Transportation Research Part B: Methodological*, 36(9):755 – 778, 2002. ISSN 0191-2615. doi: 10.1016/S0191-2615(01)00027-3. URL <http://www.sciencedirect.com/science/article/pii/S0191261501000273>.
- [27] Lars Kjær Nielsen, Leo Kroon, and Gábor Maróti. A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research*, 220(2): 496–509, 2012. ISSN 03772217.
- [28] Marc Peeters and Leo Kroon. Circulation of railway rolling stock: a branch-and-price approach. *Computers & operations research*, 35(2):538–556, 2008. ISSN 1873765x.
- [29] Daniel Potthoff, Dennis Huisman, Daniel Potthoff, Dennis Huisman, and Guy Desaulniers. Column generation with dynamic duty selection for railway crew rescheduling. *Transportation Science*, 44(4):493–505, 2010. ISSN 15265447.
- [30] Natalia J. Rezanova and David M. Ryan. The train driver recovery problem—a set partitioning based model and solution method. *Computers & Operations Research*, 37(5):845 – 856, 2010. ISSN 0305-0548. doi: 10.1016/j.cor.2009.03.023. URL <http://www.sciencedirect.com/science/article/pii/S0305054809000938>.
- [31] David M. Ryan and Brian A. Foster. An integer programming approach to scheduling. In A. Wren, editor, *Computer Scheduling of Public Transport: Urban Passenger Vehicle and Crew Scheduling*, pages 269–280. North-Holland, 1981.
- [32] Alexander Schrijver. Minimum circulation of railway stock. *CWI Quarterly*, 6:205–217, 1993.
- [33] Koorush Ziarati, François Soumis, Jacques Desrosiers, Sylvie Gélinas, and André Saintonge. Locomotive assignment with heterogeneous consists at {CN} north america. *European Journal of Operational Research*, 97(2):281 – 292, 1997. ISSN 0377-2217. doi: [http://dx.doi.org/10.1016/S0377-2217\(96\)00198-1](http://dx.doi.org/10.1016/S0377-2217(96)00198-1). URL <http://www.sciencedirect.com/science/article/pii/S0377221796001981>.