

BRADY GILG TORSTEN KLUG
ROSEMARIE MARTIENSSEN JOSEPH PAAT
THOMAS SCHLECHTE CHRISTOF SCHULZ
SINAN SEYMEN ALEXANDER TESCH

Conflict-Free Railway Track Assignment at Depots

Zuse Institute Berlin
Takustr. 7
14195 Berlin
Germany

Telephone: +49 30-84185-0
Telefax: +49 30-84185-125

E-mail: bibliothek@zib.de
URL: <http://www.zib.de>

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Conflict-Free Railway Track Assignment at Depots

Brady Gilg¹, Torsten Klug⁶, Rosemarie Martienssen³, Joseph Paat⁴, Thomas Schlechte², Christof Schulz², Sinan Seymen⁵, and Alexander Tesch⁶

¹Arizona State University

²Dres. Löbel, Borndörfer & Weider GbR

³Goethe University Frankfurt

⁴Johns Hopkins University

⁵Sabanci University

⁶Zuse Institute Berlin

May 3, 2017

Abstract

Managing rolling stock with no passengers aboard is a critical component of railway operations. In particular, one problem is to park the rolling stock on a given set of tracks at the end of a day or service. Depending on the parking assignment, shunting may be required in order for a parked train to depart or for an incoming train to park. Given a collection of tracks \mathcal{M} and a collection of trains \mathcal{T} with fixed arrival-departure timetable, the *train assignment problem (TAP)* is to determine the maximum number of trains from \mathcal{T} that can be parked on \mathcal{M} according to the timetable and without the use of shunting. Hence, efficiently solving the TAP allows to quickly compute feasible parking schedules that do not require further shunting adjustments. Similar assignment questions where shunting is allowed have been considered in [2, 9, 13, 14]. In this paper, we present two integer programming models for solving the TAP. To our knowledge, this is the first integrated approach that considers track lengths along with the three most common types of parking tracks. We compare these models on a theoretical level. We also prove that a decision version of the TAP is NP-complete, justifying the use of integer programming techniques. Using stochastic and robust modelling techniques, both models produce parking assignments that are optimized and robust according to random train delays. We conclude with computational results for both models, observing that they perform well on real timetables.

1 Introduction

The general problem of vehicle dispatching in a depot has been widely studied in the literature and comes in many different flavors. In general, a depot consists

of a fixed number of tracks that are commonly distinguished by three different track types:

- *First-in-first-out (FIFO)*: track with one arrival and one opposite departing side (queue);
- *Last-in-first-out (LIFO)*: track with equal arrival and departure side (stack);
- *Free (FREE)*: track that can be entered and departed from both sides.

The global problem is to find a feasible assignment of vehicles to tracks such that certain arrival and departure criteria are satisfied. The problem was first mentioned in the context of tram dispatching, see Winter and Zimmermann [16] and Winter [15]. Later their work was extended by a new mathematical model of Di Gallo and Miele [10]. In the area of train unit dispatching, or train shunting respectively, Freling et al. [9] study a related model for LIFO and FREE tracks where arriving train units are assigned to tracks and possible departures. Due to the large model size they decompose the problem into two subproblems:

- 1) the assignment of arriving parts to departure parts which form a *train* and
- 2) the assignment of trains to tracks

where both problems are solved consecutively. For the more involving second problem they propose an exponential integer programming model that is solved by column generation. Kroon et al. [13] approach the integrated problem by using a compact formulation. Due to the integration of problem 1) they were not able to model FREE tracks by the strongest possible formulation, because this involves too many decision variables. Haahr et al. [12] also solve the integrated problem for LIFO tracks by a recovering approach that alternates between problems 1) and 2) where problem 2) is either solved by a compact or an exponential formulation. Moreover, Borndörfer and Cardonha [2] solve the integrated problem for FIFO tracks by a strong set partitioning model that can be solved efficiently by specific column generation and pricing methods. Differently, Bohlin et al. [1] solve the integrated problem in a multistage planning environment by a novel exponential and compact formulation that compute complete train sequences for each track. Most of the presented models focus on minimizing the number of invalid assignments, the number of shunting movements or more generic cost for practical purposes such as train type distribution and robustness. More generally, Di Stefano and Koci [7] give several complexity results concerning the assignment of trains to different track types such as FIFO and FREE tracks.

Apart from that, much research focuses on forming specific outbound train compositions from a set of individually arriving train units. For a comprehensive survey on the different variants of train shunting problems we refer to Boysen et al. [3].

Contribution. In this paper we focus on problem 2) where trains with fixed arrival and departure times are assigned to tracks. In contrast to previous approaches that consider only one or two track types at the same time, we compute track assignments for the three track types FIFO, LIFO and FREE in an integrated approach. For this, we introduce two integer programming models

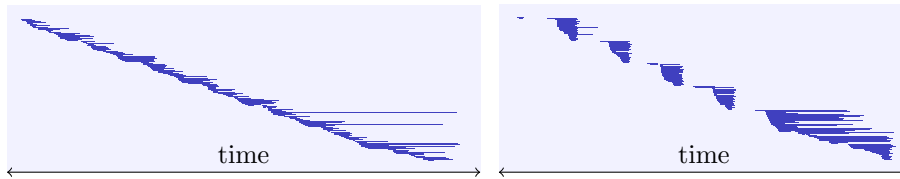


Figure 1: Different parking time interval patterns over one week.

which maximize the number of trains that can be parked without any additional shunting movement. Compared to Kroon et al. [13], we are able to model FREE tracks in its stronger version, since problem 1) is not considered. Moreover, we provide polyhedral results between our two models. Since shunting plans are very vulnerable to train delays we also present an extension of our models that further optimizes according to stochastic and robust aspects. The robustness of shunting plans has been studied mainly to restrict the number of shunting moves, for example by a recoverable robustness approach of Cicerone et al. [6] and a recovery strategy of Büsing et al. [5]. However, the impact of train delays on the planned depot schedule had been investigated rarely. In this context, we analyze the tradeoff between the possible number of parked trains without shunting conflicts and the stability of the depot schedule with respect to train delays.

The considered track assignment problem arises in rolling stock rotation planning. There fast methods are needed to verify if a given set of trains can be parked at a depot.

Outline. The article is organized as follows. In Section 2, we formally introduce the problem of maximizing the number of trains that can be parked without any shunting movement, which we denote as the *train assignment problem* (TAP). We also show the complexity of the TAP. Section 3 introduces and compares two different integer programming models for solving the TAP. Building upon this, in Section 4 we introduce methods of robustness and stochastic optimization against random delays in train arrivals. We provide extensive computational results of our models in Section 5, and finish with concluding remarks in Section 6.

2 The Train Assignment Problem

Let $\mathcal{T} = \{i_1, \dots, i_n\}$ be a collection of trains where each train $i \in \mathcal{T}$ is given a fixed arrival time $a_i \geq 0$, a fixed departure time $d_i \geq 0$ and a length $l_i > 0$. We assume that $a_i < d_i$ for all $i \in \mathcal{T}$. Moreover, we are given a set of tracks $\mathcal{M} = FF \cup LF \cup FR$ in a depot where FF , LF and FR denote the set of FIFO, LIFO and FREE tracks, respectively. In addition, every track $m \in \mathcal{M}$ is given a length $L_m > 0$.

For example, Figure 1 depicts two different instances and the considered parking time intervals of each train. There can be different patterns where trains arrive and depart continuously or all at once.

A *feasible parking assignment* is defined as an assignment of trains in \mathcal{T} to tracks

in \mathcal{M} such that

- (a) each train is assigned to at most one track;
- (b) on any track and at any time, the total length of all currently parking trains does not exceed the track length;
- (c) each train assigned to a FREE track has a designated arrival and departure side;
- (d) each train that leaves a track must not be blocked by an other train that stands between the departing train and its departure side.

The *size* of a feasible parking assignment is denoted as the number of assigned trains.

Definition 1 (The Train Assignment Problem (TAP)). *Given a set of trains \mathcal{T} and a set of tracks \mathcal{M} , find a feasible parking assignment between \mathcal{T} and \mathcal{M} that has maximum size.*

A feasible solution to the TAP is represented by a partition $\mathcal{T} = \bigcup_{m \in \mathcal{M}} \mathcal{T}_m$ where $\mathcal{T}_m \subseteq \mathcal{T}$ denotes the set of trains that are assigned to track $m \in \mathcal{M}$.

Example 1. *Assume three trains $\mathcal{T} = \{i_1, i_2, i_3\}$ with arrival- and departure times $a_1 = 0, a_2 = 1, a_3 = 2, d_1 = 4, d_2 = 5, d_3 = 3$ and lengths $l_i = 1$ for all $i \in \mathcal{T}$. Furthermore, assume two tracks $\mathcal{M} = \{T_1, T_2\}$ where T_1 is a LIFO track (stack) and T_2 is a FIFO track (queue) where both tracks have length $L_1 = L_2 = 2$.*

- *Suppose the train assignment $\mathcal{T}_1 = \{i_1, i_2\}$ and $\mathcal{T}_2 = \{i_3\}$. In this case, i_2 arrives after i_1 but also leaves after i_1 on a LIFO track. Thus, i_2 blocks the departure of i_1 and therefore the train assignment is infeasible.*
- *Now suppose the train assignment $\mathcal{T}_1 = \{i_1\}$ and $\mathcal{T}_2 = \{i_2, i_3\}$. This constitutes a feasible train assignment since properties (a)-(d) are satisfied. It is also a maximum train assignment because all trains can be parked on the given tracks.*
- *Finally, suppose an additional FREE track T_3 of length $L_3 = 3$ and the train assignment $\mathcal{T}_3 = \{i_1, i_2, i_3\}$ with $p_1 = 0, p_2 = 0, p_3 = 0, q_1 = 1, q_2 = 1, q_3 = 0$ where $p_i \in \{0, 1\}$ is the arrival side and $q_i \in \{0, 1\}$ the departure side for all $i \in \mathcal{T}$ ($0 = \text{left}, 1 = \text{right}$). This yields a feasible and maximum train assignment since all properties (a)-(d) are satisfied*

In the following we show that the TAP is NP-hard. To the best of our knowledge, this complexity result does not appear in the literature.

Definition 2 (k -Train Assignment Problem (k -TAP)). *Given a set of trains \mathcal{T} , a set of tracks \mathcal{M} , and $k \in \mathbb{N}$, does there exist a feasible parking assignment of size k ?*

The k -TAP corresponds to the decision version of the TAP. Since $k \leq n$ is polynomial in the input, solving the k -TAP for all $k = 1, \dots, n$ solves the TAP. We show that answering the k -TAP is NP-complete by using a reduction from the *partition problem*, which is well known to be NP-complete [11].

Definition 3 (Partition Problem (PP)). Let $B = \{b_1, \dots, b_n\}$ be a collection of positive integers. The partition problem is to decide whether or not there exists a subset $B' \subset B$ such that

$$\sum_{b_i \in B'} b_i = \sum_{b_i \in B \setminus B'} b_i.$$

Proposition 1. The k -TAP is NP-complete.

Proof. Given a solution to k -TAP, we can verify in polynomial time if this solution yields a 'yes' answer, since conditions (a)-(d) can be checked in polynomial time and $k \leq n$ is polynomial. Therefore, k -TAP is in NP.

We show that the partition problem is polynomially reducible to k -TAP. Consider an instance of PP with positive integers $B = \{b_1, \dots, b_n\}$. If $M = \sum_{i=1}^n b_i$ is odd, then the answer to PP is 'no'. So we may assume that M is even. We make the following reduction. For each $b_i \in B$, define a train i with length $l_i = b_i$, arrival time $a_i = i$ and departure time $d_i = n + i$. Define the train collection $\mathcal{T} = \{i \in \mathcal{T} : b_i \in B\}$ and let $\mathcal{M} = \{T_1, T_2\}$ be the set of two FIFO tracks with lengths $L_1 = L_2 = \frac{M}{2}$. Create an instance $(\mathcal{T}, \mathcal{M}, k)$ of the k -TAP with $k = n$. This construction is polynomial in n and k .

First suppose the answer to PP is 'yes' for $B' \subset B$ with $\sum_{b_i \in B'} b_i = \sum_{b_i \in B \setminus B'} b_i = \frac{M}{2}$. The train collections $\mathcal{T}_1 = \{i \in \mathcal{T} : b_i \in B'\}$ and $\mathcal{T}_2 = \{i \in \mathcal{T} : b_i \in B \setminus B'\}$ yield a feasible parking assignment to tracks T_1 and T_2 respectively. Hence the answer to the corresponding k -TAP is also 'yes' and can be retrieved in time that is polynomial in n and k .

Now suppose that the answer to the k -TAP is 'yes' and define $B' = \{b_i \in B : i \text{ parks on } T_1\}$. At time n , all trains park simultaneously on tracks T_1 and T_2 . Since any feasible parking assignment satisfies the track length restriction, we have $\sum_{b_i \in B'} b_i = \sum_{b_i \in B \setminus B'} b_i = \frac{M}{2}$, otherwise one track length is exceeded. Hence, the answer to PP is also 'yes' and can be retrieved in time that is polynomial in n and k .

Consequently, the answer to PP is 'yes' if and only if the answer to k -TAP is 'yes' and all transformations can be performed in polynomial time. This completes the proof. \square

3 Integer Programming Models for the TAP

We provide two integer programming models for solving the TAP. The two models share the same set of decision variables and agree on certain constraints. In fact, they differ in the handling of assignment conflicts for different track types.

3.1 Model preliminaries

FIFO and LIFO tracks have a unique side to enter and leave the track. Thus, for each tuple $(i, m) \in \mathcal{T} \times (FF \cup LF)$, define the decision variable $x_{im} \in \{0, 1\}$ with $x_{im} = 1$ if train i is assigned to track m . In contrast, FREE tracks can be entered and exited from two sides. We label one side by '0' (left) and the other side by '1' (right). For each 4-tuple $(i, m, p, q) \in \mathcal{T} \times FR \times \{0, 1\} \times \{0, 1\}$,

define the decision variable $x_{i,m,p,q} \in \{0, 1\}$ with $x_{i,m,p,q} = 1$ if train i enters track m from side p and leaves from side q . This explicit modeling of the FREE tracks was proposed but not used in Kroon et al. [13], since they consider a larger problem.

In the TAP, the number of assigned trains is maximized. Hence, the objective function can be stated as

$$\max \sum_{i,m} x_{i,m} + \sum_{i,m,p,q} x_{i,m,p,q}. \quad (1)$$

The basic model considers the following set of constraints

$$\sum_m x_{i,m} + \sum_{m,p,q} x_{i,m,p,q} \leq 1 \quad \forall i \in \mathcal{T} \quad (2)$$

$$\sum_{\substack{j \in \mathcal{T}: \\ a_j \leq a_i < d_j}} l_j x_{j,m} \leq L_m \quad \forall i \in \mathcal{T}, m \in FF \cup LF \quad (3)$$

$$\sum_{\substack{j \in \mathcal{T}: \\ a_j \leq a_i < d_j}} \sum_{p,q \in \{0,1\}} l_j x_{j,m,p,q} \leq L_m \quad \forall i \in \mathcal{T}, m \in FR. \quad (4)$$

Inequalities (2) say that each train can be assigned to at most one track. Furthermore, inequalities (3) and (4) indicate that for each arriving train on some track, the length of all currently parking trains must not exceed the track length.

3.2 The departure model

The idea of the departure model is to focus on the departure time of one train on a given track and to identify possible arrival time relations to other currently parking trains.

First suppose a train $i \in \mathcal{T}$ parks on a FIFO track $m \in FF$. Then any train that arrived before i must not be on the track at time d_i . Hence any train $j \in \mathcal{T}$ that is on track m at time d_i must have arrived after i . That is, $a_j < d_i < d_j$ and $x_{i,m} = x_{j,m} = 1$ implies $a_i \leq a_j$, what is modeled by

$$a_i x_{i,m} - a_i(1 - x_{j,m}) \leq a_j x_{j,m} \quad \forall i, j \in \mathcal{T} : a_j < d_i < d_j, m \in FF. \quad (5)$$

Now consider a LIFO track $m \in LF$ and suppose train $i \in \mathcal{T}$ parks on m . Then there cannot be any train on track m at time d_i that arrived after i . Hence any train $j \in \mathcal{T}$ on track m at time d_i must have arrived before i . That is, $a_j < d_i < d_j$ and $x_{i,m} = x_{j,m} = 1$ implies $a_j \leq a_i$, what is modeled by

$$a_j x_{j,m} - a_j(1 - x_{i,m}) \leq a_i x_{i,m} \quad \forall i, j \in \mathcal{T} : a_j < d_i < d_j, m \in LF. \quad (6)$$

Finally, consider a free track $m \in FR$ and suppose train $i \in \mathcal{T}$ parks on m . We consider two different cases based on the arrival and departure times of i . In the first case, assume that train i is assigned to arrive and depart from side p of track m . In this case, i uses m as a LIFO track. Thus, similar to the LIFO case, there must not exist a train $j \in \mathcal{T}$ on track m at time d_i that arrived after i from side p . That is, if $a_j < d_i < d_j$ holds and trains i and j park on track m

where j arrived from side p then $a_j \leq a_i$ must hold, what is modeled by

$$a_j \sum_q x_{j,m,p,q} - a_j(1 - x_{i,m,p,p}) \leq a_i x_{i,m,p,p} \quad (7)$$

$$\forall i, j \in \mathcal{T} : a_j < d_i < d_j, m \in FR, p \in \{0, 1\}.$$

In the second case, suppose train i arrives from side p and departs from side $q \neq p$ of track m . In this situation, i uses m as a FIFO track. Hence, any train $j \in \mathcal{T}$ that is parked on m at time d_i must have entered from side p not earlier than a_i . Therefore, if $a_j < d_i < d_j$ holds and trains i and j park on track m where j arrives from side p then $a_i \leq a_j$ must hold, what is modeled by

$$a_i x_{i,m,p,q} - a_i(1 - \sum_{p',q'} x_{j,m,p',q'}) \leq a_j \sum_{q'} x_{j,m,p,q'} \quad (8)$$

$$\forall i, j \in \mathcal{T} : a_j < d_i < d_j, m \in FR, p \neq q \in \{0, 1\}.$$

With these constraints, the *departure model* becomes

$$\max \sum_{i,m} x_{i,m} + \sum_{i,m,p,q} x_{i,m,p,q} \quad (1)$$

subject to

$$\text{Each train assigned to at most one track} \quad (2)$$

$$\text{Length constraints} \quad (3)-(4)$$

$$\text{Departure constraints} \quad (5)-(8)$$

Decision variables are binary

3.3 The conflict model

In the conflict model, we focus on forbidding infeasible assignments of two trains.

Let $i, j \in \mathcal{T}$ be two trains with $a_i < a_j$. In this case, the first set of inequalities for the conflict model is given by

$$x_{i,m} + x_{j,m} \leq 1 \quad \forall i, j \in \mathcal{T} : a_i < a_j < d_i < d_j, m \in FF \quad (9)$$

$$x_{i,m} + x_{j,m} \leq 1 \quad \forall i, j \in \mathcal{T} : a_i < a_j < d_j < d_i, m \in LF \quad (10)$$

$$\sum_{p,q} x_{i,m,p,q} + x_{j,m,0,1} + x_{j,m,1,0} \leq 1 \quad \forall i, j \in \mathcal{T} : a_i < a_j < d_j < d_i, m \in FR \quad (11)$$

$$\sum_p x_{i,m,p,0} + \sum_q x_{j,m,0,q} \leq 1 \quad \forall i, j \in \mathcal{T} : a_i < a_j < d_i < d_j, m \in FR \quad (12)$$

$$\sum_p x_{i,m,p,1} + \sum_q x_{j,m,1,q} \leq 1 \quad \forall i, j \in \mathcal{T} : a_i < a_j < d_i < d_j, m \in FR \quad (13)$$

which can be described as follows. An assignment of i and j to a FIFO track is infeasible if $d_i < d_j$, because j blocks the departure of i . Similarly, an assignment to a LIFO track is infeasible if $d_j < d_i$, because i blocks the departure of j . Those conflicts are modeled by constraints (9) and (10). For the FREE tracks,

we distinguish two cases. If $d_j < d_i$ holds then the departure of j is blocked by i , if j passes the track either from side 0 to 1 or from side 1 to 0. This is expressed by constraints (11). Otherwise, if $d_i < d_j$ holds then i is blocked by j , if i departs from the side where j arrives. This is given by the symmetric inequalities (12) and (13). Since we allow simultaneous departures, no conflict is imposed for $d_i = d_j$.

Now, let $i, j \in \mathcal{T}$ be two trains with $a_i = a_j$ and $d_i < d_j$. For LIFO and FIFO tracks, we suppose that a_i or a_j can be shifted by a small value such that exactly no conflict occurs. For the FREE tracks, consider the set of inequalities

$$x_{i,m,0,1} + \sum_q x_{j,m,1,q} \leq 1 \quad \forall i, j \in \mathcal{T} : a_i = a_j < d_i < d_j, m \in FR \quad (14)$$

$$x_{i,m,1,0} + \sum_q x_{j,m,0,q} \leq 1 \quad \forall i, j \in \mathcal{T} : a_i = a_j < d_i < d_j, m \in FR \quad (15)$$

that express the following. If $d_i < d_j$ holds, then i is blocked by j , if i and j arrive from opposite sides and i passes the track either from side 0 to 1 or from side 1 to 0. This is given by constraints (14) and (15).

Finally, consider the case with $a_i = a_j$ and $d_i = d_j$ for two trains $i, j \in \mathcal{T}$ which implies the inequalities

$$x_{i,m,0,1} + x_{j,m,1,0} \leq 1 \quad \forall i, j \in \mathcal{T} : a_i = a_j < d_i = d_j, m \in FR \quad (16)$$

$$x_{i,m,1,0} + x_{j,m,0,1} \leq 1 \quad \forall i, j \in \mathcal{T} : a_i = a_j < d_i = d_j, m \in FR \quad (17)$$

that forbid i and j to arrive at opposite sides, and for each to pass the track either from side 0 to 1 or from side 1 to 0.

When added to the assignment constraints (2), the length constraints (3)-(4), the binary constraints, and the conflict constraints (9)-(17) precisely describe the set of feasible parking assignments. The *conflict model* is defined as

$$\max \sum_{i,m} x_{i,m} + \sum_{i,m,p,q} x_{i,m,p,q} \quad (1)$$

subject to

$$\text{Each train assigned to at most one track} \quad (2)$$

$$\text{Length constraints} \quad (3)-(4)$$

$$\text{Conflict constraints} \quad (9)-(17)$$

Decision variables are binary

3.4 Comparing the departure and conflict models

The departure model and the conflict model share the same set of variables. In total, there are $N = |\mathcal{T}| \cdot (|LF| + |FF| + 4|FR|)$ many variables. The number

of constraints in both models is dominated by their individual constraints of Sections 3.2 and 3.3. There every pair of trains implies at most one constraint on every track, so the number of constraints is $O(|\mathcal{T}|^2|\mathcal{M}|)$ in both models.

Although both model sizes are of the same magnitude, it is of general interest to examine the theoretical quality of the two models, for example in terms of the strength of their LP-relaxations. Hence, define

$$P_D = \{x \in [0, 1]^N \mid x \text{ satisfies (2), (3), (4), (5) - (8)}\} \quad (18)$$

$$P_C = \{x \in [0, 1]^N \mid x \text{ satisfies (2), (3), (4), (9) - (17)}\} \quad (19)$$

as the induced polytopes of the LP-relaxations of the departure model (P_D) and the conflict model (P_C).

Theorem 1. *For every instance of the TAP, we have $P_C \subset P_D$. Thus the conflict model has the stronger LP-relaxation.*

Proof. Consider a TAP instance with a set of trains \mathcal{T} and tracks \mathcal{M} . We show that the inequalities (5)-(8) defining P_D are dominated by inequalities defining P_C . We consider each inequality separately.

- 1) Rewrite inequalities (5) as $a_i x_{i,m} + (a_i - a_j) x_{j,m} \leq a_i$. If $a_j < a_i$ the inequality is dominated by the conflict inequality $a_i x_{i,m} + a_i x_{j,m} \leq a_i$ (9), otherwise it is trivially satisfied.
- 2) Rewrite inequalities (6) as $a_j x_{j,m} + (a_j - a_i) x_{i,m} \leq a_j$. If $a_j < a_i$ the inequality is dominated by the conflict inequality $a_j x_{j,m} + a_j x_{i,m} \leq a_j$ (12),(13), otherwise it is trivially satisfied.
- 3) Rewrite inequalities (7) as $a_j \sum_{q'} x_{j,m,p,q'} + (a_j - a_i) x_{i,m,p,p} \leq a_j$. If $a_j > a_i$ this inequality is dominated by the conflict inequality $a_j \sum_{q'} x_{j,m,p,q'} + a_j \sum_{p'} x_{i,m,p',p} \leq a_j$ (10), otherwise it is trivially satisfied.
- 4) Rewrite inequalities (8) as $a_i x_{i,m,p,q} + (a_i - a_j) \sum_{p',q'} x_{i,m,p',q'} \leq a_i$. If $a_j < a_i$ holds the inequality is dominated by the conflict inequality $a_i x_{i,m,p,q} + a_i \sum_{p',q'} x_{i,m,p',q'} \leq a_i$ (9), otherwise it is trivially satisfied.

□

Proposition 2. *There are instances of the TAP where the conflict model is strictly tighter than the departure model. That is, there are instances satisfying $P_C \subsetneq P_D$.*

Proof. Let $\mathcal{M} = \{T_1\}$ be the track set where T_1 is a FIFO track with length $L_1 = 2$. Let $\mathcal{T} = \{i_1, i_2\}$ be the train set, where $a_1 = 1, a_2 = 2, d_2 = 3, d_1 = 4$ and $l_1 = l_2 = 1$. Since we have two trains and only one track, the only two decision variables are x_{11} and x_{21} and so $P_D, P_C \subseteq \mathbb{R}^2$. We use the notation that x_{11} represents the first coordinate and x_{21} represents the second. With this, we have $P_C = \text{conv}\{(0, 0), (1, 0), (0, 1)\}$ and $P_D = \text{conv}\{(0, 0), (0, 1), (1, \frac{1}{2}), (1, 0)\}$, see Figure 2.

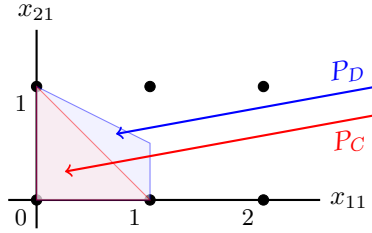


Figure 2: The polytope P_C can be strictly contained in the polytope P_D .

□

Proposition 2 shows that the linear relaxation of the conflict model can be strictly weaker than the linear relaxation of the departure model.

4 Stochastic and Robust Optimization Against Delays

Solving the TAP using the departure or conflict model results in a feasible parking assignment. In practice, however, trains are subject to delays and the computed TAP solution may become infeasible. Therefore, it is necessary to compute schedules that are robust against individual delays. In this section, we propose a *stochastic* and a *robust* extension of our models.

4.1 Modeling delays

Suppose two trains $i, j \in \mathcal{T}$ with planned arrival and departure times $a_i \leq a_j < d_i < d_j$ are assigned to a FIFO track. This assignment is valid. Now suppose that train i is delayed and arrives after time a_j . Now, this represents a conflict on a FIFO track and the previous assignment becomes invalid, see (9). Symmetrically, delays may turn a feasible LIFO assignment to a LIFO conflict, see (10). Therefore, we penalize for every pair of trains $i, j \in \mathcal{T}$ with $a_i \leq a_j$ the probability that i arrives later than j by an additional cost term in the objective function. For this, we suppose that train delays follow an *exponential distribution* which is a natural assumption, see [4].

The density function of the exponential distribution with parameter $\lambda > 0$ is defined as

$$f_\lambda(t) = \begin{cases} \lambda e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (20)$$

and its cumulative distribution function is defined as

$$P(D \leq t) = F_\lambda(t) = \int_{-\infty}^t f_\lambda(\theta) d\theta = \begin{cases} 1 - e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (21)$$

where $D \sim Exp(\lambda)$ is an exponentially distributed random variable with rate parameter $\lambda > 0$. In particular, $F_\lambda(t)$ denotes the probability that the delay is not larger than t time units. For each train $i \in \mathcal{T}$, define the stochastically

independent random variable $D_i \sim Exp(\lambda_i)$ that models the arrival time delay of train i with decay rate $\lambda_i > 0$. For large values of λ_i it is more likely that train i arrives on time. From the common expression for the difference of two exponentially distributed random variables, we derive the coefficients

$$\begin{aligned}\pi_{ij} = P(a_i + D_i > a_j + D_j) &= 1 - P(D_i - D_j \leq a_j - a_i) \\ &= \frac{\lambda_j}{\lambda_i + \lambda_j} \cdot e^{-\lambda_i(a_j - a_i)}\end{aligned}$$

for all train pairs $i, j \in \mathcal{T}$ with $a_i \leq a_j$. The values $\pi_{ij} \in (0, 1)$ express the probability that train i arrives later than train j . The coefficients π_{ij} are added as additional cost coefficients to the objective function to penalize an assignment of trains $i, j \in \mathcal{T}$ with $a_i \leq a_j$ to the same track.

4.2 Model extensions

Let $\mathcal{T}_a = \{(i, j) \in \mathcal{T} \times \mathcal{T} \mid a_i < a_j \vee (a_i = a_j \wedge i < j)\}$ be the set of consecutively arriving train pairs.

4.2.1 Stochastic

For any two trains $(i, j) \in \mathcal{T}_a$, define the decision variable $y_{i,j} \in \{0, 1\}$ with $y_{i,j} = 1$, if trains i and j are assigned on the same track. This relation is given by the inequalities

$$x_{i,m} + x_{j,m} \leq 1 + y_{i,j} \quad \forall (i, j) \in \mathcal{T}_a, m \in FF \cup LF \quad (22)$$

$$\sum_{p,q} (x_{i,m,p,q} + x_{j,m,p,q}) \leq 1 + y_{i,j} \quad \forall (i, j) \in \mathcal{T}_a, m \in FR \quad (23)$$

that are added additionally to either the departure model or the conflict model. Let $\gamma \in [0, 1]$ be a weight parameter and define the stochastic objective function as

$$\max \gamma \left(\sum_{i,m} x_{i,m} + \sum_{i,m,p,q} x_{i,m,p,q} \right) - (1 - \gamma) \sum_{(i,j) \in \mathcal{T}_a} \pi_{ij} \cdot y_{i,j} \quad (24)$$

which additionally minimizes the average number of assignment conflicts caused by delays. This leads to a multi-objective optimization problem. Therefore, our model is evaluated for different values of γ .

4.2.2 Robust

In the robust case, we consider a threshold probability $\tilde{\pi} \in (0, 1)$ up to which the assignment of two trains to the same track is considered as secure. Hence, if $\pi_{ij} \leq 1 - \tilde{\pi}$ holds for two trains $(i, j) \in \mathcal{T}_a$ then i and j are allowed to be assigned to the same track. Otherwise, if $\pi_{ij} > 1 - \tilde{\pi}$ we forbid such assignments. Therefore, we add the inequalities

$$x_{i,m} + x_{j,m} \leq 1 \quad \forall (i, j) \in \mathcal{T}_a : \pi_{ij} > 1 - \tilde{\pi}, m \in FF \cup LF \quad (25)$$

$$\sum_{p,q} (x_{i,m,p,q} + x_{j,m,p,q}) \leq 1 \quad \forall (i, j) \in \mathcal{T}_a : \pi_{ij} > 1 - \tilde{\pi}, m \in FR. \quad (26)$$

Consequently, to make the train assignment more robust against delays we have to choose $\tilde{\pi}$ as large as possible. In turn, this may reduce the number of assignable trains. Therefore, the computation for different values of $\tilde{\pi}$ reveals how much train units, or shunting movements respectively, it costs to enforce a specific planning reliability.

5 Computational Results

As basis of our computational results, we used real-world train timetable data of regional passenger traffic in Italy and Sweden. Our test set consists of 12 instances where each represents a standard week for one depot with individual numbers of trains, track types and properties, see Table 1. The column *frequency* denotes the general frequency of the arriving trains which ranges from *single, uniform* to *rush*. The column *duration* represents the general parking duration of the trains where *overnight* means that most trains arrive at one day, park over night and leave the next day. Finally, the column *weekend* indicates, if most trains that arrive on Fridays or Saturdays will park during the weekend (*yes/no/both equally*). For each train we are given the arrival and departure times at the depot and its train length. Moreover, for each depot we are given the number of different track types and their respective lengths. We distinguish three test cases: *deterministic*, *stochastic* and *robust*. In the deterministic case, the departure model and the conflict model are evaluated according to their computational performance on the test set, no delays are considered. The stochastic and the robust case include exponentially distributed train delays, as introduced in Section 4. We used rate parameters $\lambda_i = 1$ for all trains $i \in \mathcal{T}$ which gives a mean exponential delay of one hour. In the stochastic case, a weighted objective between the number of assigned trains and the expected number of delay conflicts is optimized. In the robust case, the number of assigned trains is maximized with respect to fixed upper bounds on the probability that two trains induce a delay conflict.

The proposed mixed-integer programming models have been solved with the commercial solver Gurobi 6.5. All our computations were performed on a desktop computer with 32 GB of RAM and an Intel Xeon CPU E3-1245v3 with four cores @3.40 GHz.

Table 1: Information on the instances

Instance	$ \mathcal{T} $	$ FR $	$ LF $	$ FF $	frequency	duration	weekend
instance 1	203	1	1	1	uniform	overnight	yes
instance 2	61	0	0	1	single	< 1 day	no
instance 3	328	1	1	2	rush	1-2 days	yes
instance 4	171	1	1	0	uniform	overnight	no
instance 5	152	2	1	0	uniform	overnight	both
instance 6	198	0	3	0	single	1-2 days	both
instance 7	187	0	4	0	rush	overnight	yes
instance 8	101	2	2	2	rush	overnight	no
instance 9	84	0	5	0	rush	overnight	yes
instance 10	237	4	5	3	rush	overnight	yes
instance 11	90	2	3	0	uniform	overnight	yes
instance 12	176	14	0	0	rush	overnight	yes

5.1 Deterministic

In the deterministic case, we run the departure model and the conflict model on the given test instances, compare Table 2. As stated, both models have the same number of variables. Practically, the departure model has about twice as many constraints as the conflict model. Both models were able to solve all instances to optimality and most of the instances could be solved within a few seconds. However, there is one hard instance with many trains that took 42 minutes to be solved in the conflict model and 73 minutes in the departure model. But the main effort was spent on proving optimality, since after a few seconds we obtained solutions within 5% of optimality. We conclude that the practical performance of the two models reflects the proven theoretical strength, that is the conflict model outperforms the departure model. Therefore, we use the conflict model for the robust and stochastic computations.

From a practical point of view, every test case admits an optimal solution where almost every train can be assigned in a conflict-free manner. In the planning process, this helps to identify a set of generally hard assignable trains that can either be handled individually or treated as good start solution for further optimization. For example, a two-level approach where the remaining trains are integrated by a heuristic or a more complex optimization algorithm to minimize shunting movements seems promising.

Table 2: Comparing the running times of the conflict and departure model.

Instance	\mathcal{T}	opt	DM			CM		
			# rows	# cols	runtime	# rows	# cols	runtime
instance 1	203	203	4974	1218	0.05	2535	1218	0.03
instance 2	61	53	135	61	0.00	135	61	0.00
instance 3	28	326	21315	2296	4355.18	11496	2296	2499.06
instance 4	71	170	7688	855	0.15	2952	855	0.08
instance 5	52	152	13199	1368	0.37	5221	1368	0.17
instance 6	98	176	8526	594	0.78	4467	594	0.67
instance 7	87	181	8811	748	0.21	4375	748	0.14
instance 8	01	101	7377	1212	0.13	3985	1212	0.06
instance 9	84	82	2499	420	0.02	1389	420	0.02
instance 10	237	237	97596	5688	3.39	44516	5688	1.50
instance 11	90	89	7360	990	0.25	3234	990	0.15
instance 12	76	176	118392	9856	3.97	42988	9856	1.50

5.2 Stochastic

In the stochastic model, we compute solutions according to a weighted objective between the total number of assigned trains and the expected number of stochastic assignment conflicts. For practical reasons, we only define variables $y_{i,j}$ with $\pi_{i,j} \geq 0.001$, otherwise the model includes too many variables that almost do not affect the objective function.

Each instance is computed for the weights $\gamma \in \{0.1, 0.2, 0.3, 0.4, 0.9\}$, where small values of γ indicate a higher penalty for the expected number of conflicts. Notably, the stochastic case is significantly harder to solve than the deterministic case since constraints (22) and (23) represent weak linear inequalities that yield poor LP-relaxations. The instances become even harder to solve for small values of γ , that is when the stochastic objective gets a higher weight. However, the obtained primal solutions seem reasonable even though the dual bounds are very weak on some instances.

In particular, our solutions reflect the competing behaviour between the two objectives very well, see Table 3. For each instance and each weighting parameter γ the table contains the percentage of assigned trains compared to the maximum number of assignable trains and the expected number of assignment conflicts that are caused by train delays. As anticipated, small values of γ yield

solutions with less expected conflicts but only few assigned trains. Conversely, high values of γ yield more assigned trains but also more expected conflicts caused by delays. In addition, we tested the quality of exactly these solutions by generating random scenarios according to the assumed exponential distribution of delays. Table 4 shows the average number of assignment conflicts over all scenarios. One can see that the transition between the numbers of delay conflicts is smooth for increasing γ values. However, Table 4 shows a in a slightly higher number of conflicts than Table 3. This is because in each scenario the delay of one specific train may induce several conflicts to other trains while in the stochastic optimization approach the expected conflicts are treated independently of each other. In general, one would like to compute all pareto-optimal points of this multi-criteria objective, for example by the *weighted sum method* [8], which we leave for future research. However, even the computation of almost pareto-optimal solutions is useful because it provides a broad portfolio of different solutions according to concurrent objectives. In the end, we can simply choose the best compromise solution in the portfolio that respects both objectives in the best possible way.

Table 3: Stochastic results for different weighting parameters γ . The table lists the percentage $\Sigma x\%$ of assigned trains compared to the maximum number of assignable trains and the expected number of assignment conflicts $\Sigma\pi$.

Instance	γ									
	0.1		0.2		0.3		0.4		0.9	
	$\Sigma x\%$	$\Sigma\pi$	$\Sigma x\%$	$\Sigma\pi$	$\Sigma x\%$	$\Sigma\pi$	$\Sigma x\%$	$\Sigma\pi$	$\Sigma x\%$	$\Sigma\pi$
instance 1	65.0	2.8	79.8	7.9	91.1	15.7	99.0	24.3	100.0	24.9
instance 2	66.0	0.7	77.4	2.2	92.5	4.6	100.0	6.5	100.0	6.3
instance 3	47.2	2.8	56.4	7.9	70.9	23.4	84.0	46.2	99.7	90.1
instance 4	30.6	1.2	33.5	2.3	45.9	9.1	55.3	18.2	100.0	107.2
instance 5	42.1	0.6	55.9	4.5	63.8	8.3	71.7	15.3	100.0	62.1
instance 6	34.7	1.3	39.2	2.8	43.8	5.0	61.9	22.5	100.0	116.2
instance 7	53.6	0.7	69.6	5.8	71.8	6.9	82.9	17.8	100.0	58.4
instance 8	93.1	0.9	99.0	1.8	100.0	2.2	100.0	2.2	100.0	2.1
instance 9	87.8	0.6	96.3	1.8	97.6	2.0	100.0	2.9	100.0	2.8
instance 10	84.0	1.8	92.8	5.1	97.9	8.7	100.0	10.8	100.0	9.6
instance 11	87.6	0.6	96.6	1.9	98.9	2.6	100.0	3.1	100.0	3.0
instance 12	100.0	0.6	100.0	0.6	100.0	0.5	100.0	0.5	100.0	0.4

Table 4: The average number of assignment conflicts over 10000 scenarios for the obtained stochastic solutions with different weighting parameter γ .

instance	γ								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
instance 1	1.19	3.06	5.72	9.40	11.08	10.96	10.48	10.93	11.06
instance 2	0.13	0.33	1.04	1.65	1.65	1.65	1.65	1.65	1.65
instance 3	1.69	5.01	14.60	29.34	45.72	50.97	55.80	54.46	53.28
instance 4	0.40	0.43	3.85	7.10	13.45	22.04	24.71	25.47	25.41
instance 5	0.26	1.88	3.17	6.54	13.29	18.04	19.49	17.63	20.28
instance 6	0.70	1.27	2.34	7.61	11.07	16.76	25.51	32.45	39.43
instance 7	0.68	5.32	6.42	10.75	16.54	17.68	26.63	40.64	40.65
instance 8	0.75	1.67	2.00	2.05	2.02	2.01	2.05	1.99	1.98
instance 9	0.29	0.78	1.06	1.63	1.63	1.63	1.63	1.63	1.63
instance 10	1.24	3.39	5.39	6.10	6.50	6.29	7.43	7.02	7.03
instance 11	0.39	1.28	1.42	2.08	2.08	2.13	2.14	2.14	2.14
instance 12	0.23	0.14	0.12	0.20	0.17	0.18	0.27	0.29	0.27

5.3 Robust

The robust model is computed for different values $\tilde{\pi} \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99\}$ in order to examine how the degree of robustness affects the number of efficiently assignable trains. High values of $\tilde{\pi}$ indicate a higher robustness against delays. For instance, $\tilde{\pi} = 0.99$ requires that for every pair of trains the probability that their planned arrival order changes is not greater than 1%. In this case, however, the total number of assignable trains decreases.

For each of the test instances we obtained optimal solutions that did not take a considerably higher amount of computation time than the deterministic case. A first positive aspect of this method is that we get an idea how much capacity is needed by a depot to ensure a certain amount of robustness, see Table 5. In every instance, an optimal assignment is achieved for $\tilde{\pi} = 0.5$. Surprisingly, even small changes to $\tilde{\pi} = 0.6$ can lead to drastic decreases in the number of assignable trains, see instance 4 for example. This is caused by a comparably huge number of simultaneously arriving trains. In such cases, the assignment plan is very sensitive to delays and therefore robustness is hard to achieve in general. Furthermore, we tested the practical behaviour of the computed train assignments by randomly generated scenarios where the number

of occurring conflicts is counted, see Table 6. One can observe that a robust train assignment significantly reduces the expected number of delay conflicts. However, this is achieved only with a smaller number of assigned trains. Hence, the robust solutions can be seen as an indicator for the cost of planning certainty. Moreover, it can be used as robust start solution in a two-level approach, as suggested in the deterministic case.

Table 5: Percentage of the number of assigned trains compared to the maximum possible number (*opt*) of assignable trains for the optimal solution with robustness value $\tilde{\pi}$.

instance	opt	$\tilde{\pi}$						
		0.99	0.95	0.9	0.8	0.7	0.6	0.5
instance 1	203	46.8	64.0	76.8	91.6	95.6	98.0	100.0
instance 2	53	39.6	64.2	66.0	96.2	96.2	100.0	100.0
instance 3	326	34.4	46.6	58.3	75.2	89.3	97.2	100.0
instance 4	170	17.1	25.3	30.6	33.5	45.9	45.9	100.0
instance 5	152	32.2	42.1	44.1	56.6	63.8	66.4	100.0
instance 6	176	25.0	26.1	35.8	43.2	47.7	51.1	100.0
instance 7	181	45.3	50.8	58.0	72.9	72.9	75.1	100.0
instance 8	101	77.2	91.1	98.0	100.0	100.0	100.0	100.0
instance 9	82	76.8	86.6	92.7	98.8	98.8	100.0	100.0
instance 10	237	70.9	80.6	89.0	98.3	99.6	100.0	100.0
instance 11	89	77.5	85.4	91.0	97.8	98.9	100.0	100.0
instance 12	176	92.0	100.0	100.0	100.0	100.0	100.0	100.0

Table 6: The average number of assignment conflicts over 10000 scenarios for the optimal solution with robustness parameter $\tilde{\pi}$.

instance	opt	$\tilde{\pi}$						
		0.99	0.95	0.9	0.8	0.7	0.6	0.5
instance 1	203	0.17	1.43	2.33	7.58	9.72	11.45	20.17
instance 2	53	0.02	0.26	0.19	2.18	2.18	1.67	1.67
instance 3	326	0.23	2.01	5.58	17.87	37.77	58.01	85.16
instance 4	170	0.06	0.35	0.55	0.53	3.09	3.09	41.21
instance 5	152	0.25	0.44	0.55	2.08	3.17	4.53	33.37
instance 6	176	0.05	0.16	0.86	2.66	4.48	6.22	54.93
instance 7	181	0.12	0.52	2.28	8.72	9.19	12.81	47.40
instance 8	101	0.16	0.71	2.21	3.99	6.07	7.05	8.85
instance 9	82	0.06	0.41	0.81	1.82	2.25	3.36	3.54
instance 10	237	0.30	1.21	3.13	8.70	12.88	15.25	21.58
instance 11	89	0.06	0.43	0.85	2.04	2.59	4.75	5.71
instance 12	176	0.10	0.28	0.70	1.00	1.34	2.78	3.22

6 Conclusion

In this article we introduced the Train Assignment Problem (TAP) and proposed two integer programming models for the TAP to compute conflict-free assignments of trains to tracks in a depot. Our models include the three most common track types FIFO, LIFO and FREE within an integrated approach. Furthermore, our models have been extended to a stochastic and robust optimization approach against train delays. The obtained results show that the proposed models are suited well to compute exact solutions to the TAP. Since in particular the manual planning of FREE tracks can be very challenging, our models can notably improve the efficiency of a depot planner. Although it is already efficient in practice, it seems beneficial to study further polyhedral properties of the conflict model.

If train delays are considered, our models help to estimate and to optimize the stability of the train assignment. Our stochastic model computes a list of compromise solutions between the two competing objectives of finding a maximal shunting-free schedule and to avoid delay conflicts. Hence, it supports the planner to balance these two objectives. Moreover, our robust approach is well suited to compute assignments that are generally stable against delay conflicts. This gives an estimate for the cost of robustness of an instance.

In fact, our model can still be improved in some aspects. Since the number of conflict-free trains is maximized, one can develop methods that assign the remaining trains under individual preferences (e.g. minimize shunting). This can be done, for example, by fast heuristics or by a more sophisticated two-level approach that is built on top. Our model may also operate within an alternating recovering approach, similar to [12]. Several directions seem plausible. Apart from that, the estimation of train delays can be improved. Although the exponential distribution is quite common in the literature, in certain cases it does not reflect the reality very well since punctual arrivals and long delays seem too unlikely. Therefore, more accurate probability distributions, preferably in combination with historical data, would further improve the applicability of our models.

The main motivation of our model is to function as a subsolver in a more comprehensive framework to compute rolling stock rotation plans. In this context, it works as a feasibility check to quickly decide if a given set of trains can be parked in the depot. This allows to include or exclude train rotations that occur during the computations. For this particular problem, our model already serves its purpose.

Moreover, we see the possibility to enhance the proposed model for the task of designing a depot. To the best of the authors knowledge there are no scientific studies which tackles this particular task. Infrastructure decisions like the construction or expansion of a depot or a station layout are basically political and strategic decisions. Thus, it might be interesting to extend the mathematical models in order to demonstrate that the capacity provided by a depot is sufficient for given expected operations.

Acknowledgements

This work has been developed within the **Research Campus MODAL** (Mathematical Optimization and Data Analysis Laboratories) funded by the German Ministry of Education and Research (BMBF). This research was performed while some authors were participating in a program of the Institute for Pure and Applied Mathematics (**IPAM**), which is supported by the National Science Foundation.

References

- [1] Markus Bohlin, Sara Gestrelus, Florian Dahms, Matúš Mihalák, and Holger Flier. Optimization methods for multistage freight train formation. *Transportation Science*, 2015.
- [2] Ralf Borndörfer and Carlos Cardonha. A set partitioning approach to shunting. *V Latin American Algorithms, Graphs, and Optimization Symposium*, 160(18):2636–2644, 2012.
- [3] Nils Boysen, Malte Flidner, Florian Jaehn, and Erwin Pesch. Shunting yard operations: Theoretical aspects and applications. *European Journal of Operational Research*, 220(1):1–14, 2012.
- [4] Keith Briggs and Christian Beck. Modelling train delays with q-exponential functions. *Physica A: Statistical Mechanics and its Applications*, 378(2):498–504, 2007.
- [5] Christina Büsing and Jens Maue. Robust algorithms for sorting railway cars. In *European Symposium on Algorithms*, pages 350–361. Springer, 2010.
- [6] Serafino Cicerone, Gianlorenzo D’Angelo, Gabriele Di Stefano, Daniele Frigioni, and Alfredo Navarra. 12. robust algorithms and price of robustness in shunting problems. In *OASICs-Open Access Series in Informatics*, volume 7. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2007.
- [7] Gabriele Di Stefano and Magnus Love Koči. A graph theoretical approach to the shunting problem. *Electronic Notes in Theoretical Computer Science*, 92:16–33, 2004.
- [8] Matthias Ehrgott. *Multicriteria optimization*. Springer Science & Business Media, 2006.
- [9] Richard Freling, Ramon M Lentink, Leo G Kroon, and Dennis Huisman. Shunting of passenger train units in a railway station. *Transportation Science*, 39(2):261–272, 2005.
- [10] Giorgio Gallo and Federico Di Miele. Dispatching buses in parking depots. *Transportation Science*, 35(3):322–330, 2001.
- [11] Michael R. Garey and David S. Johnson. *Computers and Intractability. A guide to the Theory of NP-Completeness*. W.H. Freeman, New York, 1997.

- [12] Jørgen Thorlund Haahr, Richard Martin Lusby, Jesper Larsen, and David Pisinger. Simultaneously recovering rolling stock schedules and depot plans under disruption. In *13th Conference on Advanced Systems in Public Transport*, 2015.
- [13] Leo G Kroon, Ramon M Lentink, and Alexander Schrijver. Shunting of passenger train units: An integrated approach. *Transportation Science*, 2008.
- [14] Gabriele Di Stefano and Mahnus Love Koči. A graph theoretical approach to the shunting problem. *Electronic Notes in Theoretical Computer Science*, 92:16–33, 2004.
- [15] Thomas Winter. *Online and Real-Time Dispatching Problems*. PhD thesis, Technischen Universität Braunschweig, 1999.
- [16] Thomas Winter and Uwe T Zimmermann. Real-time dispatch of trams in storage yards. *Annals of Operations Research*, 96(1-4):287–315, 2000.