Journal of Rail Transport Planning & Management 14 (2020) 100198

Contents lists available at ScienceDirect

Journal of Rail Transport Planning & Management

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

A dynamic model for real-time track assignment at railway yards

B.B.W. Schasfoort, K. Gkiotsalitis *, O.A.L. Eikenbroek, E.C. van Berkum

University of Twente, Center for Transport Studies, Horst - Ring Z-222, P.O. Box 217, 7500 AE Enschede, The Netherlands

ARTICLE INFO

Keywords:
Track assignment
Real-time control
Rail operations
Minimizing delays
Rescheduling

ABSTRACT

In this paper, we study the real-time train assignment problem (RT-TAP) that arises from the unreliable arrival times of freight trains and the last-minute parking requests at railway yards. In the RT-TAP, the reassignment of trains to the yard is triggered every time a train arrives at the railway yard and needs to be assigned (event-based optimization). After introducing a problem formulation for the RT-TAP problem, we prove that RT-TAP is NP-Hard. In particular, the RT-TAP is modeled as a mixed integer program that strives to minimize the total weighted delay of trains. Because of its computational complexity and the time-critical nature of this problem, we introduce two real-time solution methods: (a) a problem-specific genetic algorithm (GA), (b) and a first-scheduled first-served (FSFS) heuristic. In small instances, we show that the GA returns a globally optimal solution which is identical to the solution of exact optimization methods. In larger problem instances, the heuristic approaches of FSFS and GA are tested at the Waalhaven Zuid railway yard in the Netherlands using two months of operational data. In the experimental results, the GA solutions reduce the average delays by more than 4 min compared to the solutions of the FSFS heuristic.

1. Introduction

Since the use of the first railway in the early 19th century, rail freight transportation has always played an important role in the transportation sector (Network Rail, 2016). As discussed by Givoni and Banister (2013), after long and short haul maritime transport, rail freight transport is the mode with the least CO_2 emissions per kg/tonne-km.

While the rail transportation sector is still growing, the increase in demand could cause capacity problems on the network. For example, US transportation officials already stated their concerns that future capacity limitations of the American railway system are likely to result in a degradation of speed and reliability of the network (U.S. Railroad, 2008). While a need for further expansion of railway networks is strongly desired around the globe, the most common solution would be an expansion of the railway network by constructing new infrastructure. However, constructing new infrastructure is generally decided by policymakers and involves a long-term decision process requiring large investments (Narayanaswami and Rangaraj, 2011; Boysen et al., 2012). Because of high construction costs, it is more efficient to increase the occupancy and transit rate on the network in order to comply with the increase of traffic.

To comply with the demand for rail freight transporters, timetables of trains are determined one year in advance depending on the country. When a train is driving according to its schedule, the automatic route setting (ARS) system automatically assigns a train to a track (Pachl, 2004). Ideally, a train dispatcher (TD), who is responsible for the movement of trains in an assigned area (for example a railway yard), only needs to intervene when unplanned events occur. As discussed by Piner and Condry (2017), there is a possibility that unplanned events result in disturbances and/or disruptions. Now, when a delayed freight train arrives at a railway

* Corresponding author.
E-mail address: k.gkiotsalitis@utwente.nl (K. Gkiotsalitis).

https://doi.org/10.1016/j.jrtpm.2020.100198
Received 26 July 2019; Received in revised form 13 March 2020; Accepted 28 March 2020
Available online 3 April 2020

Table 1
Delays of freight trains: originally planned compared to executed data at Waalhaven Zuid.

	Too early (in min)			On time	Too late (in min)			
	+60	60–30	30–5	−5 to +5	5–30	30–60	60–120	+120
Number of trains	16	8	25	85	40	18	28	40
Percentage (%)	6%	3%	10%	33%	15%	7%	11%	15%
Totals:	19% too early			33%	48% too late			

Table 2
Executed trains from predetermined plan at Waalhaven Zuid.

	Year plan	Original plan	Scheduled plan	Executed
Number of trains	389	515	573	573
Percentage (%)	66%	90%	100%	100%

yard, the track needs to be assigned manually by the TD (D'Ariano, 2015; Josyula et al., 2018). Because of the large complexity of finding an optimal assignment of trains to tracks, the manual planning of delayed trains in real time might result in large optimality gaps, especially when inexperienced TDs are involved (U.S. Department of Transportation, 1999).

When considering the arrival times of rail freight trains in Europe, it appears that 67% arrive within a 15-minute time frame of their original schedule. Furthermore, 19% of the trains were more than 3 h delayed, and 4% arrived after 24 h or their original schedule (European Commission, 2014). Concerning the Waalhaven Zuid yard in the Netherlands, historical data shows that only 33% of the trains arrive within a ± 5-minute time frame of their originally planned time at their destination yard. 48% of the trains arrive more than 5 min later than originally scheduled, whereas 26% and 15% of these trains arrive more than one or two hours late, respectively (see Table 1).

Large deviations in arrivals are likely to be related to the fact that rail freight is a part of a supply chain that also includes other modes such as barges and trucks. If one of these modes is delayed, it might cause a knock-on delay on the train that is waiting for its cargo. This could mean that the freight train leaves the yard already with a delay. Furthermore, when focusing on the distance traveled by freight trains, most of them are likely to cross borders of different countries resulting in increased travel time uncertainty.

When we consider the performance of freight trains on any railway network, we also need to consider the other rail users. This means that if a train is delayed, it needs to drive in-between other trains without influencing their schedules. The complexity of re-arranging the delayed train arises because it has to perform on the mixed network including the different characteristics of other trains such as speed, acceleration, deceleration, and cargo. When a train is delayed, it can be temporarily stored on a side track so that trains that drive on time can be accommodated. This partly explains the large percentage of trains with more than two or three hours of delay (Timmermans, 2018). As substantiated by Briggs and Beck (2007), delays typically follow an exponential distribution. Because of the high percentage and the high variance of delays, changing the assignment of freight trains at the execution phase is not a trivial task and is therefore related to the experience and adequate handling of the TD.

To add further complexity to the assignment process, high fluctuation on demand from the client side results in a late notice of train requesting paths. For example, Table 2 shows that:

- 66% of the executed trains are planned one year in advance (*yearly plan*);
- 90% of the executed trains are known three days before execution, when making the *original plan*;
- and 10% of the trains make a last-minute request to park at the yard (*scheduled plan*).

The large delays on the network and the high percentage of last-minute requests contribute to the complexity of assigning trains to tracks in real-time, further referred to as the RT-TAP problem. If a suitable fit is not possible, inbound trains might need to wait or be rerouted to another railway yard, which can result in larger delays. In addition, the large delays of trains, and the high percentage of trains scheduled with a small prior notice underscores the number of manual decisions made at railway yards.

To tackle this issue, this study models the RT-TAP and investigates real-time mathematical optimization techniques that aim to minimize the total delays of outbound trains at a railway yard. The input of the model is the set of tracks and trains. The set of tracks includes the track number and its corresponding length. The set of trains includes the length of the train, its planned arrival and departure time, the time that the train needs to spend at the yard, and a train weighting. The train weighting is related to the train's schedule in relation to the schedule of other trains, the type of train, or the type of cargo (U.S. Department of Transportation, 1999). The output of the model provides an optimal assignment solution of trains to tracks that minimizes the total weighted delay on outbound trains from the railway yard. This assignment is updated every time a train arrives at the yard triggering an event-based re-optimization. The TD can make the final decision based on our proposed assignment.

The remainder of this study is structured as follows. In Section 2 we provide a summary of the relevant literature. After this, we identify the research gap, explain the research goal, and discuss the main contributions of this paper with respect to the state-of-the-art. Section 3 describes the overall framework and introduces the mathematical program of the RT-TAP. This program can be solved in small to medium scale scenarios using exact optimization methods because RT-TAP is proved to be NP-Hard. In Section 4, we discuss two different solution methods for solving the RT-TAP. First, we introduce a problem-specific GA, and then a FSFS heuristic solution method. In Section 5, we perform numerical experiments in smaller instances to determine the optimality gap(s) of the two heuristics. Furthermore, we apply our method in Waalhaven Zuid railway yard located in the western part of the Netherlands using historical data from the 1st of October until the 1st of December, 2015. Finally, we conclude our work and provide future research directions in Section 6.

2. Related studies

This section aims to describe the current literature in the field of railway operations with regards to train rescheduling and rerouting at the tactical and operational level.

2.1. Train rescheduling

In track assignment problems, most works consider sorting and scheduling problems at marshaling or shunting yards (Hansmann and Zimmermann, 2008). Shunting can be described as the process where the train composition changes on successive train services (Haahr and Lusby, 2017; Gatto et al., 2009; Boysen et al., 2012). Jaehn et al. (2018, 2015) describe the assignment problems including shunting operations as the railcar retrieval problem (RRT). With this problem, each freight car receives a priority value which is linked with the due date of each outbound train. The main objective of this problem is to minimize the total shunting operation costs by minimizing the total weighted departure of all outbound trains at a shunting yard. Gestrelius et al. (2017) developed an integer programming model for scheduling shunting tasks as well as allocating arrival yard tracks and classification bowl tracks. Haahr et al. (2017) described the train unit shunting problem (TUSP), which entails assigning train units from a depot or shunting yard to a scheduled train service in such a way that a conflict in the resulting operations is avoided. An important constraint from the TUSP is that all tracks must be processed in last-in first-out (LIFO) order, which means that trains can only enter from one side of the yard.

The track assignment problem (TAP) determines the maximum number of trains that can be assigned to a yard without the use of shunting. Gilg et al. (2018) shows that the TAP is NP-hard and presents two integer programming models for solving this problem. The approach integrates track lengths along with the three most common types of yard layouts: first-in first-out (FIFO), LIFO, and FREE tracks, where FREE is a combination of both LIFO and FIFO. Kang et al. (2019) solve an offline track assignment problem considering the stochasticity in the system and focusing on detecting limited carrying capacity areas at railway stations.

Minimizing delays is a multidisciplinary problem that is widely discussed in many different fields (Schöbel, 2007; Schachtebeck and Schöbel, 2008, 2010). A survey on problem models and solution approaches with regards to the rescheduling in railway networks is developed by Fang (2015). Several works such as Dollevoet et al. (2011) and Schöbel (2009) discuss minimizing the total (weighted) delays of trains, and use fast-to-compute heuristics to solve such problems (Dollevoet and Huisman, 2014).

Because the models presented above focus at the tactical planning level, arrival times of trains can be modeled by including a stochastic and robust extension in order to consider the arrival time uncertainties in the optimization process (Gilg et al., 2018; Boysen et al., 2012; Briggs and Beck, 2007).

2.2. Real-time algorithms for rescheduling railway systems

In the previous section we discussed different track assignment problems; however, most of them refer to the tactical level and cannot reschedule the track assignment in case of disruptions during the actual operations. To rectify this, Cai and Goh (1994) proposed to reassign trains to tracks in real-time. Past works mostly consist of real-time optimization models for solving railway networks when disruptions occur (Bettinelli et al., 2017). An overview of recovery models and algorithms for real-time railway rescheduling was presented by Cacchiani et al. (2014).

As discussed by Dollevoet et al. (2010), railway disruption management is a combination of three different aspects: timetables, rolling stock and crew. The real-time rescheduling of long-distance, high-speed trains in a highly disrupted situation is discussed by Zhan et al. (2015). Zhan et al. (2015) developed a decision support system (DSS) which found solutions for real-time rescheduling within 10 min of computational time. Other models developed by Fischetti and Monaci (2017) and Corman et al. (2010) found practical solutions for real-time train rescheduling within seconds. Winter and Zimmermann (2000) developed a heuristic approach that assigns trams to tracks in real-time considering the departure of the remaining trams of the day. If a global optimum is not found in real-time, trams can be reassigned again afterwards to comply with the schedule of each tram for the next day.

The works described above all present models solved in real-time. The time limitation to solve a problem in real-time is not strongly specified. For example, while Fischetti and Monaci (2017) and Corman et al. (2010) tried to find practical solutions within seconds, Zhan et al. (2015) accepted finding solutions within 10 min. In our study, the computational time limit for solving the RT-TAP is equal to the time between the last time we receive an accurate update of the expected arrival time of a train and its actual arrival at the yard. This allows the train dispatcher to suggest a track to the incoming train upon its arrival.

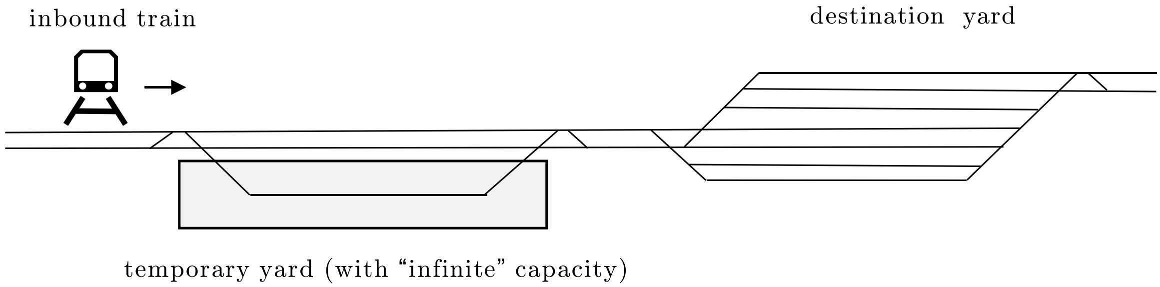

Fig. 1. Railway yard.

2.3. Contribution

From the literature review of past works, we can conclude that the area of assigning delayed trains to tracks in real-time is still not fully explored. This is also supported by Gilg et al. (2018) who indicates the lack of literature with regards to the impact of a train's delay on the planned depot schedules.

From the aforementioned studies, the works of Gilg et al. (2018) and Winter and Zimmermann (2000) are the closest prior arts to our work. However, the model of Gilg et al. (2018) refers to the tactical level and considers the schedule as fixed. It includes an expected arrival time deviation, but in actual operations the arrival and departure times can still vary from their expected values. The model presented by Winter and Zimmermann (2000) is closer to a solution of the RT-TAP as it considers trams to be parked at a yard and includes the arrival and departure times in real-time. However, the model is developed on a single stud yard and focuses on the parking, and not the transit at a yard. Also, when assignments are not ‘optimal’, it considers the tram to easily exit the current track and rearrange a track assignment. Especially the latter should be avoided in the case of freight trains because of their significant length and weight. This makes the solution algorithms presented by Gilg et al. (2018) and Winter and Zimmermann (2000) not suitable for the RT-TAP considered in this study.

In this work, we investigate the RT-TAP and provide a mathematical approach for solving the problem at hand. Furthermore, we introduce two solution methods for allocating freight trains to tracks at any railway yard in real-time. Our proposed model reassigns inbound trains to tracks every time a train arrives at the yard to minimize the total weighted delays of outbound trains in case a delay occurs.

3. Problem formulation

This section formally introduces a static version RT-TAP as a mathematical optimization problem (Section 3.1), and briefly discusses the complexity of the static problem in Section 3.2. This static problem is solved every time a train arrives at the yard to reassign trains to tracks. Hence, the number of reassignments during a day of operations is equal to the total number of daily trains that arrive at the yard.

3.1. Problem formulation

We consider the TAP, in which n trains should be assigned to m tracks at the yard, so that the total delay (with respect to the communicated/initial departure time) is minimized. At the destination yard, each train needs to undergo some operation(s) and must therefore be parked at the yard. In the next model, we make some necessary assumptions; (i) there is unlimited storage at a temporary yard that can temporarily accommodate trains which cannot be assigned at the main yard, (ii) trains do not need to be shunted, (iii) there exists a feasible path to and from the yard, (iv) each train is assigned to one and only one track and cannot be pre-empted, and (v) each track is assigned at most one train, which does not exceed the track length. Fig. 1 illustrates the conditions in the railway yard when an inbound train approaches.

We formulate an integer program (IP), well-suited for scheduling and routing problems. We consider a destination yard with a set of tracks $K = \{1, 2, \dots, m\}$, and a set of trains $N = \{1, \dots, n\}$. Each train $i \in N$ with length l_i has an initial (or: planned) arrival time a_i, initial (or: communicated) departure time d_i and processing time p_i in order to execute all operations. The length of track k is denoted by L_k.

Our formulation is a multi-commodity flow problem (*cf.* Cordeau et al. (2007)). We introduce graph $G = (V, E)$ with $V = \{0, 1, \dots, n, n+1\}$ where nodes 0 and $n+1$ are *dummy trains* that correspond to the begin and end of a schedule respectively, i.e., a schedule for a fixed track is a path from 0 to $n+1$. We assume that $E = V \times V$ (i.e., there are no precedence constraints with respect to the trains).

Let us introduce the variables. Binary variable $x_{ij}^k \in \{0, 1\}$ denotes whether (dummy) train i is scheduled before (dummy) train j on track k. Variable y_i^k, $i \in N$, $k \in K$, is the time that service of train i is started on track k. For convenience, we introduce (auxiliary) variable z_i^k as the finishing time of train i on track k. $\bar{d}_i$ denotes the departure time of train i. Before proceeding to our mathematical program, we summarize the nomenclature (see Table 3).

Table 3
Nomenclature.

Sets	
$K = \{1, 2, \dots, m\}$	Set of tracks;
$N = \{1, 2, \dots, n\}$	Set of trains;
Indices	
i	Index of trains;
k	Index of tracks;
Parameters	
l_i	Length of train i
a_i	Planned arrival time of train i
p_i	Processing time of train i at the yard to execute all operations
d_i	Initially planned (communicated) departure time of train i
L_k	Length of track k
w_i	Weight related to the delay of train i
Variables	
x_{ij}^k	Dummy 0–1 variable denoting whether train i is scheduled before (dummy) train j on track k
y_i^k	The time that service of train i is started on track k
z_i^k	Finishing time of train i on track k
$\bar{d}_i$	Actual departure time of train i

The optimization problem TAP is formulated in Eqs.(1)–(11). Note that the dynamic RT-TAP problem solves TAP every time a new train needs to be assigned to a track using the expected arrival times of incoming trains as problem parameters. Hence, RT-TAP performs a reassignment every time a train is arriving at the yard by solving the following model:

$$\min \quad \sum_{i \in N} w_i(\bar{d}_i - d_i) \tag{1}$$

subject to:

$$\sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1, \qquad i \in N \tag{2}$$

$$\sum_{j \in V} x_{0j}^k = 1, \qquad k \in K \tag{3}$$

$$\sum_{i \in V} x_{i,n+1}^k = 1, \qquad k \in K \tag{4}$$

$$\sum_{i \in V} x_{ij}^k - \sum_{i \in V} x_{ji}^k = 0, \qquad k \in K, j \in N \tag{5}$$

$$y_i^k \geq a_i, \qquad k \in K, i \in V \tag{6}$$

$$z_i^k = y_i^k + p_i, \qquad k \in K, i \in N \tag{7}$$

$$z_j^k \geq z_i^k + H - M(1 - x_{ij}^k), \qquad k \in K, (i,j) \in E \tag{8}$$

$$\bar{d}_i \geq z_i^k, \qquad k \in K, i \in N \tag{9}$$

$$\bar{d}_i \geq d_i, \qquad i \in N \tag{10}$$

$$l_i \sum_{j \in V} x_{ij}^k \leq L_k, \qquad k \in K, i \in N \tag{11}$$

Constraint (2) assures that each train i is only assigned once to a track. Constraints (3) and (4) assure that a schedule starts at (dummy) node 0 and ends at dummy train $n+1$. Constraint (5) is so that for every train a (dummy) train is scheduled before and after it (on the same track). (6) ensures that we can only schedule trains after arrival. That is, a train cannot be served before it has arrived at the yard. (7) is the latent departure time of train i on track k. This variable is latent in the sense that this variable is defined for every train-track combination, although not every train is scheduled on each track. (8) says that trains can only be scheduled after that the previous train has departed, and requires a minimum headway $H \geq 0$. This constraint also assures that our schedule contains no cycles (see, e.g., Cordeau et al. (2007)). Here, M is a large constant. (11) ensures that trains can only be scheduled on tracks that have sufficient length. Since z is defined for every train-track combination, in (9) we define $\bar{d}_i$ as the actual departure time. We note that in the objective function (1) we minimize the difference $w_i(\bar{d}_i - d_i)$ (non-negative by (10) so that – implicitly – $\bar{d}_i$ is minimized, i.e., $d_i = \max_{k \in K} z_i^k$). Hence, (10) ensures that (1) does not reward trains that depart earlier than scheduled, but only trains that are not delayed. Consequently, the objective function (1) is the sum of the total weighted delay. We hereby note that train assignment to tracks is not the sole reason of delays. Trains can arrive delayed at the yard for several other reasons which are not related to the train assignment decisions. Hence, by minimizing (1) we reduce the delays caused at the yard due to inefficient train assignment, while acknowledging the existence of other forms of delay (e.g., delayed arrivals at the yard) that are not related to the assignment. This is a typical approach in TAP (see Gilg et al. (2018)) which aims to provide a conflict-free railway track assignment and considers other forms of delay as problem input.

3.2. Complexity

We discuss the complexity of TAP. We prove that (the static) TAP is NP-Hard by a polynomial-time reduction from the $(1|r_i|\sum_i C_i)$-scheduling problem, which, on its turn, has a polynomial-time reduction from 3-PARTITION (see Lenstra et al. (1977)). The $(1|r_i|\sum_i C_i)$ scheduling problem is as follows: n jobs should be scheduled on a single machine, each job i has processing time p_i and is released at time r_i. The goal is to minimize the total completion time $\sum_i C_i$.

Theorem 3.1. *$(1|r_i|\sum_i C_i)$ can be reduced in polynomial time to the RT-TAP.*

Proof. For any instance of the $(1|r_i|\sum_i C_i)$ consider the RT-TAP with graph $G = (V, E)$ with $V = \{0, 1, \dots, n, n+1\}$ and $E = V \times V$. Now $m = 1$, and $L_1 = \infty$. Also, let p_i correspond to the processing time for any job, with $a_i = r_i$. Furthermore, let $H = 0$ and $d_i = 0$, $w_i = 1$ for all i. Then, a minimum completion-time schedule with objective value v exists if and only if RT-TAP has an optimal value $\sum_i \bar{d}_i = v$. ■

The complexity of the static TAP problem shows that for computationally complicated instances (e.g., many trains with a few tracks), a fast and global optimal solution method is out of reach. This particularly applies to the real-time case, in which optimization problems need to be solved within a limited time. Therefore, in the next section, we introduce computationally faster heuristics to solve the RT-TAP.

4. Solution methods

This section introduces solution methods used to solve the RT-TAP. Given the complexity of our problem, we introduce a problem-specific genetic algorithm (GA) (Section 4.1) and a first-scheduled first-served (FSFS) (Section 4.2) algorithm. Both algorithms do not guarantee to converge to a globally optimal solution, but compute a good solution within a limited amount of time, as it is demonstrated in our numerical experiments.

4.1. Problem-specific GA

With the incremental improvements of computer performance over the last decades, we are able to solve larger and more complex combinatorial optimization problems by (implicitly) enumerating all possible solutions (e.g., using a branch-and-bound approach). However, there will always be limits to what can be done even with the fastest supercomputers. Because the RT-TAP needs to be solved in real time, one might not have the time to use such a computationally intensive approach. Especially because finding the global optimum by enumeration might need days of computation before all options are considered.

In many practical circumstances when limited time is available, i.e. real-time optimization, it might be preferable to get a quick 'reasonably good' solution rather than to wait for much longer and get a marginally better one. Heuristic solution methods aim to find good solutions to optimization problem without evaluating all possible solutions. Several heuristic approaches used for solving real-time problems or problems with an NP-Complete or NP-Hard computational complexity are discussed in literature. Heuristic approaches include greedy algorithms (Törnquist, 2010, 2012; Gkiotsalitis and Cats, 2019), hill climbing (HC) (Rahim et al.), simulated annealing (SA) (Törnquist and Persson, 2005), genetic algorithms (GA) (Holland, 1975) and tabu search (TS) (Glover, 1986). We note here that the performance of different heuristics depends on the problem at hand and there is no general rule for selecting one heuristic over another.

Our first proposed local search solution method is a problem-specific GA that might find a reasonably good solution for the RT-TAP in larger problem instances. A GA is based on Darwin's theory of evolution and tries to find a solution based on natural selection and survival of the fittest. GA considers a pool of solutions rather than a single solution at each iteration. One of the first works including a GA is the book of Holland (1975), which describes the stages of a GA. The principal stages are (1) encoding the initial population, (2) evaluating the members of the population, (3) parent selection for offspring generation, (4) crossover and (5) mutation (Gkiotsalitis et al., 2019; Gkiotsalitis and Alesiani, 2019). Important to acknowledge is that different (meta-)heuristic methods may also be used for solving this problem as discussed above.

In this subsection, we introduce our GA that is used to solve the RT-TAP in limited time. We define the *population* $S \subset \mathcal{F}$ as a (finite) subset of the discrete feasible set $\mathcal{F}$, with $\mathcal{F}$ being the finite set of all possible train-track combinations. Hence, each population member $s \in S$ is an assignment of the n trains to the m tracks. Formally,

$$s = (s_1, s_2, \dots, s_n), \tag{12}$$

with

$$s_i \in \{k \mid l_i \leq L_k\}, \quad i \in N. \tag{13}$$

Here, each *gene* s_i denotes the assignment of train i to (feasible, with respect to the length) track s_i. It should be noted that different $x^1 \neq x^2$, both satisfying (2)–(11), might actually share a population member s. Therefore, we introduce the optimization problem $L(s)$ that orders the trains for a given train-track assignment s so that the total weighted delay (1) is minimized. This corresponds (since the trains are already assigned to a track) to solving RT-TAP (1)–(11) for each k (i.e., $|K| = 1$) with $N = \{i \in N \mid s_i = k\}$. $L(s)$ is a (finite set of) optimization problems, and is solved in limited time using a variant of the bubble sort algorithm.

1 initialization: construct population $S = \{s_1, s_2, \dots, s_\pi\}$;
2 solve $L(s)$ for each $s \in S$;
3 **repeat**
4 Construct S': the $(\pi - \pi')$ survivors from S;
5 Construct π' parents subsets $P_1, P_2, \dots, P_{\pi'}$ from S' with size 2;
6 **forall the** $P_i, i = 1, 2, \dots, \pi'$, **do**
7 apply crossover and mutation operation to P_i to obtain child c_i;
8 **end**
9 Solve $L(s)$ for each $s \in \tilde{S} = \bigcup_{i=1,2,\dots,\pi'} c_i$;
10 Construct new population $S = S' \cup \tilde{S}$;
11 **until** *termination criterion is met*;

Algorithm 1: Genetic Algorithm

Algorithm 1 summarizes our GA-approach (see, e.g., Hurink (1998)).

We hereby discuss the details of our algorithm. In line 4 of Algorithm 1 we determine the $(\pi - \pi')$, $1 \le \pi' < \pi$, survivors from S. These survivors S' are the fittest members of the generation S (i.e., those with the minimum weighted delay), and ensure that good assignments are not lost during the iteration process.

In line 7 of Algorithm 1 we construct the genes of the offspring. For each gene, we either apply a (random) crossover or mutation operation. In line 7 of Algorithm 1 we apply a joint crossover and mutation operation. That is, we construct a random $\{0, 1, 2\}$-vector

$$x = (x_1, x_2, \dots, x_n).$$

For $P_i = (r, t)$, with $|P_i| = 2$ and $r, t \in S'$, *child* c has genes $c_i = r_i$, if $x_i = 1$, and $c_i = t$ if $x_i = 0$ (see, e.g., Hurink (1998), Schneider and Kirkpatrick (2006)). If $x_i = 2$, we mutate gene s_i, $i \in N$, to a random track $c_i \in \{k \mid l_i \le L_k\}$.

The model continues to construct new populations until a total weighted delay of 0 seconds is achieved, or after a predetermined number of iterations.

We briefly discuss the pre-processing phase of the GA. It is clear from the representation (12) of a solution s, that the interchange of all trains among two tracks leads to the same objective value (given that both assignments are feasible). In order to reduce the size of the solution space, we eliminated these ‘double members’ from the feasible set.

4.2. FSFS approach

A second solution method examined in this study is the first-scheduled first-served (FSFS) heuristic. FSFS assigns trains to tracks based on which train arrives first at the yard. Several papers already discussed the FSFS approach in different rescheduling problems (Schöbel, 2009; Schachtebeck and Schöbel, 2010; Gkiotsalitis and Van Berkum, 2020). In this study, the model schedules the inbound trains after sorting tracks from short to long and trains based on arrivals. The FSFS approach tries to assign a train to the shortest feasible empty track. Algorithm 2 formalizes the FSFS heuristic.

1 initialization: Let $N = \{1, 2, 3, \dots, n\}$ be the set of trains and $K = \{1, 2, 3, \dots, m\}$ be the set of tracks. Let a_i be the scheduled arrival time of train i;
2 Let $q_k = 0$ for all $k \in K$;
3 **while** $N \neq \emptyset$ **do**
4 Let $i^* \in N$ be such that $a_{i^*} = \min_{i \in N} a_i$;
5 $T = T \setminus \{i^*\}$;
6 Choose track $k^* \in K$ that corresponds to an optimal solution of (Q_{i^*}) (or (Q'_{i^*}));
7 Assign train i^* to track k^*, let $q^*_k = \max\{q_{k^*}, a_{i^*}\} + p_{i^*} + H$;
8 **end**

Algorithm 2: First-Scheduled First-Served Heuristic

In Line 6 of the Algorithm 2, we solve the following optimization program (for a given train $i^* \in N$ with corresponding arrival time a_{i^*} and length l_{i^*})

$$(Q_{i^*}) \qquad \min_{k \in K} L_k \qquad \text{s.t.} \qquad \begin{array}{l} L_k \ge l_{i^*} \\ q_k \le a_{i^*} \end{array} .$$

Optimization problem (Q_{i^*}) finds the shortest feasible empty track at time a_{i^*}. It might be that such a track is not available (i.e., all feasible tracks are occupied). In that case, an assignment is not possible and the inbound train is parked at the temporary yard. This train is then scheduled as early as possible. Formally, if (Q_{i^*}) has no feasible solution, we resort to solving

$$(Q'_{i^*}) \qquad \min_{k \in K} q_k \qquad \text{s.t.} \qquad L_k \ge l_{i^*}.$$

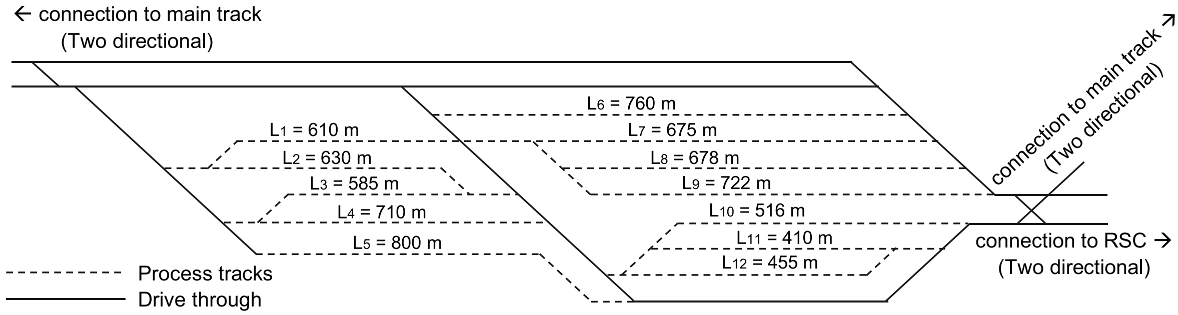

Fig. 2. Schematic overview of Whz process tracks.

5. Numerical experiments

In this section we apply our model in a numerical case study and test the performance of our solution methods. Our case study is the Waalhaven Zuid railway yard which is located in the western part of the Netherlands.

5.1. Case study description

The Waalhaven Zuid railway yard consists of about 100 tracks where each track has its own function. The functions of the tracks vary from processing, renting, passing through repair and/or parking. At Waalhaven Zuid, only the process tracks consist of electrical wiring and are centrally operated. Hence, the process tracks are the only ones that can be controlled by the train dispatcher and are considered in our case study. A schematic overview of the process tracks can be found in Fig. 2.

The characteristics of the tracks at Waalhaven Zuid can be divided into two different categories:

(1) the drive-through (solid lines);
(2) the process tracks (dashed lines).

The drive-through tracks need to remain clear at all times since they are only used for entering and exiting the yard. Instead, the 12 process tracks can be used for processing trains that enter/exit the yard.

The main purpose of trains arriving at Waalhaven Zuid is the container terminal located on the right side of the yard where trains are getting loaded and unloaded. At Waalhaven Zuid, trains first arrive from one of the two sides (dependent on the train's origin) and are then assigned to one of the 12 process tracks. After the necessary operations, the train is assigned to another track at the container terminal and the unloading and loading operations start. When the necessary operations (i.e., loading/unloading) are completed, the train continues again to one of the 12 process tracks at Waalhaven Zuid and exits the railway yard from one of the two exit tracks (dependent on the train's destination).

5.2. Scenario and input data

The data for the rail operations in Waalhaven Zuid is provided by ProRail and covers a 2-month period (from the 1st of October 2015 to the 1st of December 2015). The planning of the operations is performed at the following three levels, spanning from tactical planning to real-time control:

Original plan. (3 days in advance): Three days before execution, a so-called original plan is made where the Rail Infrastructure Manager (RIM) checks if the trains planned in the year-plan will still be executed, if there are some trains that canceled their request, or if there are additional trains that requested a path on the network.

Scheduled plan. (1.5 h in advance): A scheduled arrival of a train is communicated to the TD 1.5 h in advance. This serves as a final call to the TD because many trains arrive in the yard at a different time than expected. The TD can then determine if the train can still be assigned at the yard.

Executed plan. : The executed time is the actual time that a train arrived or departed from the yard. In the executed plan, the TD makes the final decision of the assignment of the incoming train in the yard using information from the scheduled plan with respect to the expected arrival times of incoming trains.

Evidently, our model is applied at the execution level and assigns the train to an available track given the current assignments in the yard and the expected arrival times of the incoming trains. Hence, a reassignment is performed every time a train arrives at the yard.

To investigate the performance of our model, we apply it on Thursday, October 1st — which is the busiest day of the 2-month period. The total process time needed when arriving at Waalhaven Zuid is 1h 35 min (on average). Additionally, the original plan

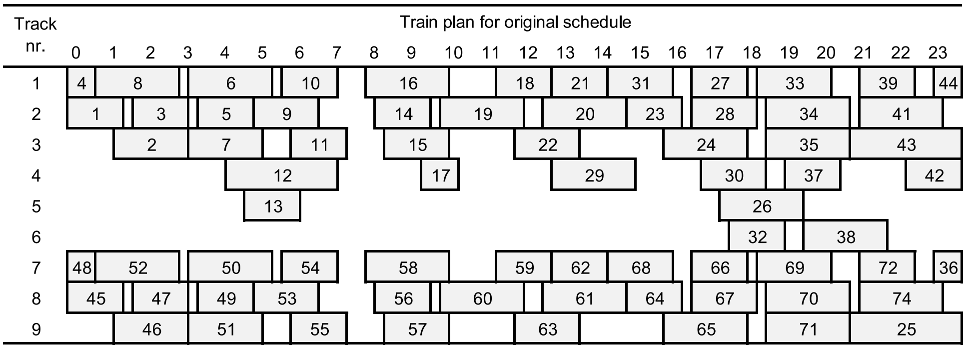

Fig. 3. Original assignment plan of trains in Waalhaven Zuid on Thursday October 1st, 2015.

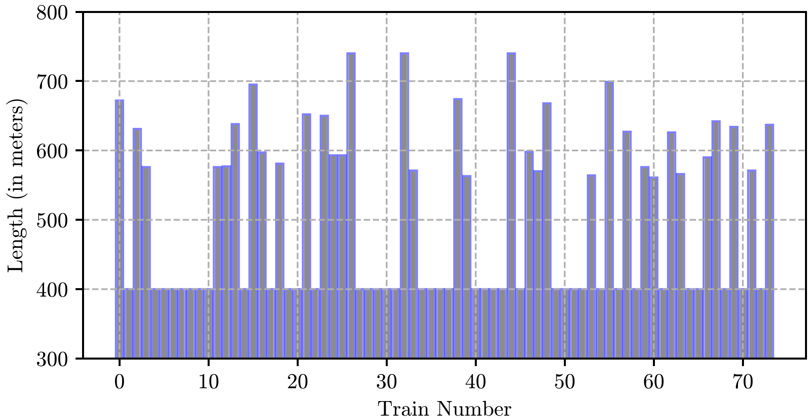

Fig. 4. Length of each train operated on the 1st of October, 2015.

of the assignment of trains is presented in Fig. 3. Note that in Fig. 3 the original schedule considers only 9 out of the 12 available process tracks in Fig. 2 because 3 of them are pre-assigned to other activities.

It is also important to note that in the Netherlands, the RIM (i.e., ProRail) is not allowed to give priority to particular trains, e.g. prioritize the assignment of one train in the expense of another. Thus, we do not use different weights for the assignment of trains in our experiments.

On the 1st of October, 2015 seventy-four (74) trains performed operations in Waalhaven Zuid. The length of each one of those trains is presented in Fig. 4. Note that trains with length of 400 m or less can be assigned at every track.

Additionally, the scheduled plan constructed by the information provided by each incoming train 1.5 h prior its arrival at the yard is presented in Fig. 5. Note that Fig. 5 presents the difference between the arrival time of a train, as it was presented in the original plan, and the scheduled arrival time, as it was communicated 1.5 h before its actual arrival.

Finally, Fig. 6 presents the differences between the arrival times of trains reported at the original plan and their actual arrival times at Waalhaven Zuid obtained from the executed plan.

5.3. Tested scenarios

To investigate the performance of the different solution methods, we compare the performance of an exact solution method against the performances of our proposed heuristic solution methods (GA and FSFS) of Section 4. The globally optimal solution of RT-TAP is found by CPLEX using the AMPL toolbox. In the remainder of this subsection, we report the optimality gap of the heuristic solution methods compared to the optimal value.

We first perform a set of experiments to test the performance of the GA and FSFS algorithm for cases in which the globally-optimal solution is known.

Our second scenario is the actual case where we consider all daily trips (74 inbound trains) that can be assigned in 9 tracks. Because CPLEX cannot perform such train assignment to tracks given the exponential increase of computational costs with the size of the problem, in this scenario we only compare the performance of our heuristics (GA and FSFS). For this, we simulate our 24-hour day including all real-time changes that occur with regards to the arrival times of trains. The input data for the larger instance is the data defined in Section 5.2.

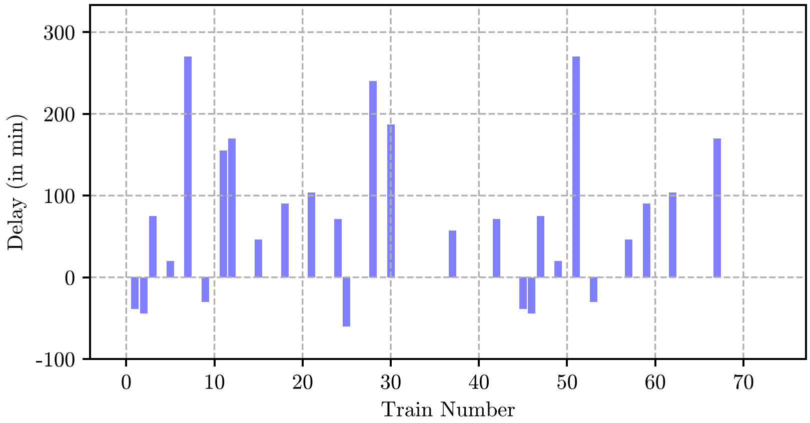

Fig. 5. Arrival time differences from the original plan to the scheduled plan on the 1st of October, 2015.

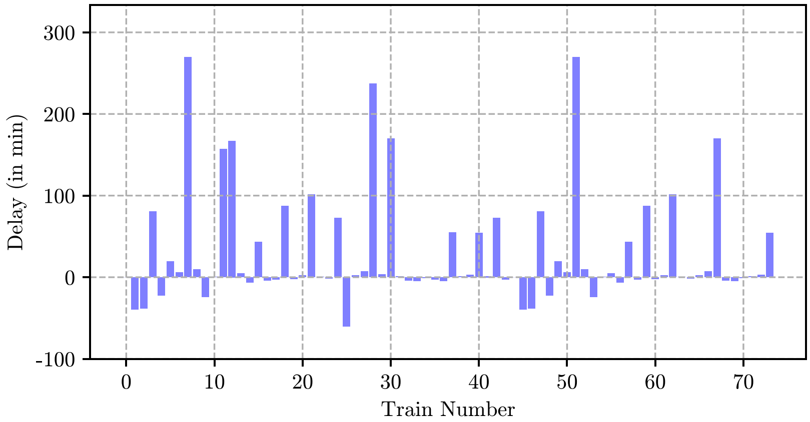

Fig. 6. Arrival time differences from the original plan to the executed plan on the 1st of October, 2015. Those differences demonstrate the realized delay of trains upon their arrival at Waalhaven Zuid.

Table 4
Hyperparameter tuning results of the GA.

Population size, π	Average optimality gap	Population generations until convergence (x10^2)
90	7.5%	270
100	1.2%	249
105	1.4%	287
110	2.4%	163

The computations are performed on a macOS High Sierra computer with a 3.4 GHz Intel Core i5 processor and 8 GB 2400 MHz DDR4 memory. In all cases, the FSFS algorithm provided solutions within a second. With regards to the performance of the GA, we measured an average computational speed of about 150 population member evaluations per second. We note that the *hyperparameters* of the GA were tuned using sensitivity analysis. In Algorithm 1 each gene of a generated offspring can either be the gene of one of its two parents, or a mutated gene (with a fixed probability of 0.02). A critical hyperparameter is the population size, π, which is tuned in hyperparameter tuning experiments. Applying our GA with different population sizes, π, in a number of artificial problem instances with more than 70 daily trips and less than 10 available tracks resulted in the average performance of Table 4. From Table 4 one can note that a population size of $\pi = 100$ results in lower optimality gaps on the average case. A population size of $\pi = 110$ allows the GA to terminate in less population generations, but results in higher optimality gaps because the GA is trapped in local optima. In all upcoming experiments in this paper, we used $\pi = 100$ and $\pi' = 90$. Additional results of this analysis with varying crossover, mutation and recombination rates are reported in Schasfoort (2019).

5.3.1. Performance on artificial instances

As discussed in Section 5.3, the performance is tested for a number of scenarios. We constructed a range of artificial instances. Each instance consists of a number of inbound trains, and a number of available tracks (see Table 5). For each instance, we compared

Table 5
Average performance of CPLEX, GA, and FSFS in different scenarios.

Tracks	Trains	Delay (s)			Run time (s)		
		CPLEX	FSFS	GA	CPLEX	FSFS	GA
2	11	27043	31452 (16.30%)	27122 (0.29%)	11	1	1
2	12	55734	64699 (16.09%)	55953 (0.39%)	1152	1	1
3	11	7311	8765 (19.89%)	7317 (0.08%)	4	1	1
3	12	6308	7335 (16.29%)	6367 (0.94%)	11	1	8
3	13	16642	19437 (16.80%)	17179 (3.23%)	197	1	25
3	14	N/A	37404 (N/A)	32071 (N/A)	>10800	1	7
4	11	4337	5688 (31.17%)	4337 (0.00%)	2	1	1
4	12	8161	10504 (28.71%)	8161 (0.00%)	26	1	19
4	13	6796	9609 (41.39%)	6907 (1.64%)	30	1	27
4	14	8760	11308 (29.09%)	9716 (10.91%)	638	1	39
5	11	336	2417 (619.04%)	336 (0.00%)	1	1	1
5	12	2880	9780 (239.62%)	2882 (0.07%)	6	1	1
5	13	4147	8703 (109.84%)	4147 (0.00%)	29	1	1
5	14	5717	17003 (197.41%)	5717 (0.00%)	233	1	23
6	11	1998	5373 (168.87%)	2137 (6.95%)	1	1	1
6	12	1222	1760 (44.06%)	1222 (0.00%)	2	1	11
6	13	2683	8091 (201.54%)	2683 (0.00%)	9	1	1
6	14	3634	6693 (84.18%)	3914 (7.71%)	111	1	1
6	15	4194	6136 (46.29%)	4194 (0.00%)	758	1	1

5 scenarios, in which we varied the length of the trains, and the corresponding arrival and departure times. In order to assess performance, we compare the average delay and the average running time over the scenarios in Table 5. We note that we could not measure running times for the FSFS and GA that were less than 1 s. The scenarios with 3 tracks and 14 inbound trains took CPLEX too much time (more than 3 h), and therefore we cannot report on the performance of the globally optimal solution. Based on the globally optimal solution obtained by CPLEX, we report the optimality gap of FSFS and the GA heuristics in columns 4 and 5 of Table 5.

When comparing the GA and FSFS algorithms with the optimal value function of the globally optimal solution calculated by CPLEX, we can conclude that the GA provides the best solution in this context considering both delay and calculation time. In particular, the calculation time of CPLEX might explode for some instances. The FSFS mechanism, on the other hand, finds a solution in very limited runtime but performs rather poorly.

Interestingly, the computation time of CPLEX can increase when the number of tracks increases considering the same number of trains. For instance, CPLEX was not able to solve the case of 3 tracks and 14 trains, whereas it solved the case of 4 tracks and 14 trains in 683 s. Its computation time reduces further to 233 s and 111 s in the cases of 5 and 6 tracks, respectively. The reason behind this is that, although the solution space increases with the increase of tracks, it is typically easier for CPLEX to find a solution when there are more available track options.

5.3.2. Performance after applying our model in our entire case study

In this sub-section we perform two experiments using the actual arrival times of trains at the yard. In the first experiment, when a train arrives at the yard it is assigned to a track according to the solution of the GA. In the second experiment, it is assigned at the yard according to the solution of the FSFS and this procedure continues until the day ends and all 74 trains have been assigned, resulting in 74 reassignments. The results in terms of total delays of trains when using the suggested assignments of the GA and the FSFS are presented in Fig. 7.

From Fig. 7, one can note that at the beginning of the day a delay starts to occur. This delay disappears after 7am because the trains start to arrive at the yard according to their original plan. The delays start again after 4pm and continue until the end of the day. A more detailed reporting of delays is provided in Table 6.

Additionally, one can observe that the GA reassignments perform better than the FSFS reassignments at almost all times of the day when delays occur. At the end of the day, the GA reassignments improved the average delay by 4 min and 42 s compared to the reassignments of the FSFS.

6. Conclusion

This work developed a mathematical model that can assist the train dispatcher in finding optimal track assignments for each inbound train considering the expected arrival times of the incoming trains in the yard and the current assignment conditions.

Our proposed model is a mixed integer program and is proved to be NP-hard by a polynomial-time reduction from the $(1|r_i|\sum_i C_i)$-scheduling problem. Because its computational intractability in large problem instances, we proposed a problem-specific GA and the FSFS heuristic solution methods. After performing numerical experiments in small-scale instances where a globally optimal solution can be computed, we showed that the GA is capable of converging to the global optimum, whereas the FSFS exhibits a substantial optimality gap.

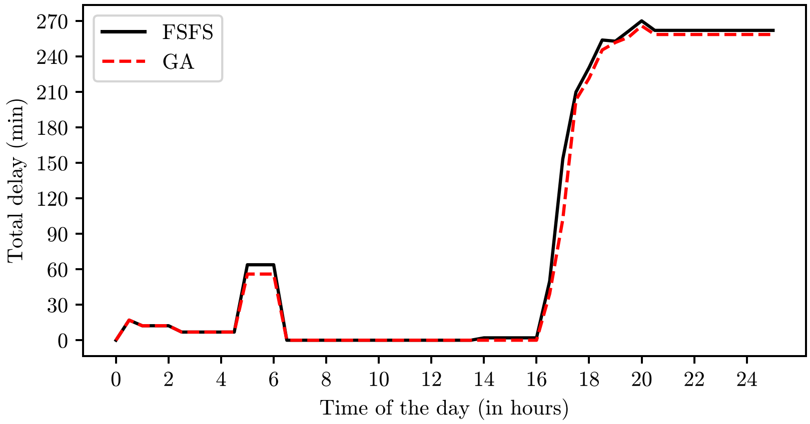

Fig. 7. Daily performance.

Table 6
Total delays of trains at different time periods of the day when reassigning trains according to the GA and FSFS solutions.

Time (hh:mm:ss)	FSFS delay (sec)	GA delay (sec)	GA improvement (sec)	Time (hh:mm:ss)	FSFS delay (sec)	GA delay (sec)	GA improvement (sec)
12:00:00 AM	735	735	0	12:30:00 PM	0	0	0
12:30:00 AM	735	735	0	1:00:00 PM	121	0	121
1:00:00 AM	735	735	0	1:30:00 PM	121	0	121
1:30:00 AM	414	414	0	2:00:00 PM	121	0	121
2:00:00 AM	414	414	0	2:30:00 PM	121	0	121
2:30:00 AM	414	414	0	3:00:00 PM	121	0	121
3:00:00 AM	414	414	0	3:30:00 PM	3000	2360	640
3:30:00 AM	414	414	0	4:00:00 PM	9200	6140	3060
4:00:00 AM	3830	3358	472	4:30:00 PM	12595	12195	400
4:30:00 AM	3830	3358	472	5:00:00 PM	13839	13296	543
5:00:00 AM	3830	3358	472	5:30:00 PM	15236	14732	504
5:30:00 AM	0	0	0	6:00:00 PM	15176	15110	66
6:00:00 AM	0	0	0	6:30:00 PM	15684	15376	308
6:30:00 AM	0	0	0	7:00:00 PM	16208	15953	255
7:00:00 AM	0	0	0	7:30:00 PM	15725	15516	209
7:30:00 AM	0	0	0	8:00:00 PM	15725	15516	209
8:00:00 AM	0	0	0	8:30:00 PM	15725	15516	209
8:30:00 AM	0	0	0	9:00:00 PM	15725	15516	209
9:00:00 AM	0	0	0	9:30:00 PM	15725	15516	209
9:30:00 AM	0	0	0	10:00:00 PM	15725	15516	209
10:00:00 AM	0	0	0	10:30:00 PM	15725	15516	209
10:30:00 AM	0	0	0	11:00:00 PM	15725	15516	209
11:00:00 AM	0	0	0	11:30:00 PM	15725	15516	209
11:30:00 AM	0	0	0	12:00:00 AM	15725	15516	209
12:00:00 PM	0	0	0				

Application of our approach to 1 day of operations in the Waalhaven Zuid yard showed that the reassignments produced by the GA algorithm result in reduced delays by 4 min and 42 s (on average) compared to the assignments of the FSFS method. As demonstrated in this study, the employment of heuristics can reduce significantly the computational costs and enable the reassignment of trains to tracks in near real-time. Overall, GA exhibited a better performance in terms of convergence and is a favorable option.

In future research, one can consider additional operations at the yard (such as shunting) and add further objectives to our model (i.e., mitigate the assignment modifications from the originally planned assignment).

Acknowledgment

The third author (O.E.) is supported by the Netherlands Organisation for Scientific Research (NWO), project number 439.16.103 (ADAPTATION).

References

Bettinelli, A., Santini, A., Vigo, D., 2017. A real-time conflict solution algorithm for the train rescheduling problem. Transp. Res. B 106, 237–265.

Boysen, N., Fiedner, M., Jaehn, F., Pesch, E., 2012. Shunting yard operations: Theoretical aspects and applications. European J. Oper. Res. 220 (1), 1–14.

Briggs, N., Beck, C., 2007. Modeling train delays with q-exponential functions. Physica A 378 (2), 498–504.

Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L., Wagenaar, J., 2014. An overview of recovery models and algorithms for real-time railway rescheduling. Transp. Res. B 63, 15–37.

Cai, X., Goh, C., 1994. A fast heuristic for the train scheduling problem. Comput. Oper. Res. 21 (5), 499–510.

Cordeau, J.-F., Laporte, G., Savelsbergh, M.W., Vigo, D., 2007. Vehicle routing. Handbooks Oper. Res. Manag. Sci. 14, 367–428.

Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M., 2010. A tabu search algorithm for rerouting trains during rail operations. Transp. Res. B 44, 175–192.

D'Ariano, A., 2015. Improving Real-Time Train Dispatching: Models, Algorithms and Applications (Ph.D. thesis). https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=2ahUKEwjhr4e-3JviAhUSDuwKHTImBHIQFjABegQIAhAC&url=https%3A%2F%2Frepository.tudelft.nl%2Fislandora%2Fobject%2Fuuid%3A178b886e-d6c8-4d39-be5d-03d9fa3a680f%2Fdatastream%2FOBJ%2Fdownload&usg=AOvVaw3PKpXVxJTtJ-NtvKMWgkWx. Accessed on 14 May 2019.

Dollevoet, T., Huisman, D., 2014. Fast heuristics for delay management with passenger rerouting. Publ. Transp. (ISSN: 1613-7159) 6 (1), 67–84.

Dollevoet, T., Huisman, D., Kroon, L.G., Veelenturf, L.P., Wagenaar, J.C., 2010. Application of an iterative framework for real-time railway rescheduling. Comput. Oper. Res. 78, 203–217.

Dollevoet, T., Schmidt, M., Schöbel, A., 2011. Delay management including capacities of stations. In: ATMOS.

European Commission, 2014. Fourth report on monitoring development of the rail market. [PDF]. https://eur-lex.europa.eu/resource.html?uri=cellar:d261b1f8-f5f4-11e3-831f-01aa75ed71a1.0002.01/DOC_2&format=PDF. Accessed on 14 May 2019.

Fang, W., 2015. A survey on problem models and solution approaches to rescheduling in railway networks. IEEE Trans. Intell. Transp. Syst. 16 (6), 2997–3016.

Fischetti, M., Monaci, M., 2017. Using a general-purpose mixed-integer linear programming solver for the practical solution of real-time train rescheduling. European J. Oper. Res. 263 (1), 258–264.

Gatto, M., Maue, J., Mihalak, M., Widmayer, P., 2009. Shunting for Dummies: An Introductory Algorithmic Survey. Springer, pp. 310–337.

Gestrelius, S., Aronsson, M., Joborn, M., Bohlin, M., 2017. Towards a comprehensive model for track allocation and rolltime scheduling at marshalling yards. J. Rail. Transp. Plan. Manag. 7, 157–170.

Gilg, B., Klug, T., Martienssen, R., Paat, J., Schlechte, T., Schulz, C., Seymen, S., Tesch, A., 2018. Conflict-free railway track assignment at depots. Rail Transp. Plan. Manag. 8, 16–28.

Givoni, M., Banister, D., 2013. Moving Towards Low Carbon Mobility. Edward Elgar.

Gkiotsalitis, K., Alesiani, F., 2019. Robust timetable optimization for bus lines subject to resource and regulatory constraints. Transp. Res. E: Logist. Transp. Rev. 128, 30–51.

Gkiotsalitis, K., Cats, O., 2019. Multi-constrained bus holding control in time windows with branch and bound and alternating minimization. Transport. B: Transport Dyn. 7 (1), 1258–1285.

Gkiotsalitis, K., Van Berkum, E., 2020. An exact method for the bus dispatching problem in rolling horizons. Transp. Res. C 110, 143–165.

Gkiotsalitis, K., Wu, Z., Cats, O., 2019. A cost-minimization model for bus fleet allocation featuring the tactical generation of short-turning and interlining options. Transp. Res. C 98, 14–36.

Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. Comput. Oper. Res. 13 (5), 533–549.

Haahr, J., Lusby, R., 2017. Integrating rolling stock scheduling with train unit. European J. Oper. Res. 259 (2), 452–468.

Haahr, J., Lusby, R., Wagenaar, J., 2017. Optimization methods for the train unit shunting problem. European J. Oper. Res. 262 (3), 981–995.

Hansmann, R.S., Zimmermann, U.T., 2008. Optimal sorting of rolling stock at hump yards. In: Krebs, H.-J., Jäger, W. (Eds.), Mathematics – Key Technology for the Future: Joint Projects Between Universities and Industry 2004–2007. Springer, Berlin, Heidelberg, ISBN: 978-3-540-77203-3, pp. 189–203.

Holland, J., 1975. Adaptation in natural and artificial systems. University of Michigan Press, Michigan.

Hurink, J., 1998. Solving Complex Optimization Problems by Local Search.

Jaehn, F., Otto, A., Seifried, K., 2018. Shunting operations at flat yards: Retrieving freight railcars from storage tracks. OR Spectrum 40 (2), 367–393.

Jaehn, F., Rieder, J., Wiehl, A., 2015. Single-stage shunting minimizing weighted departure times. Omega 52, 133–141.

Josyula, S.P., Krasemann, J.T., Lundberg, L., 2018. A parallel algorithm for train rescheduling. Transp. Res. C 95, 545–569.

Kang, L., Lu, Z., Meng, Q., 2019. Stochastic schedule–based optimization model for track allocations in large railway stations. J. Transp. Eng. A 145 (3), 04019001.

Lenstra, J.K., Kan, A.R., Brucker, P., 1977. Complexity of machine scheduling problems. In: Annals of Discrete Mathematics, vol. 1. Elsevier, pp. 343–362.

Narayanaswami, S., Rangaraj, N., 2011. Scheduling and rescheduling of railway operations: A review and expository analysis. Technol. Oper. Manag. 2 (2), 102–122.

Network Rail, 2016. Value and importance of rail freight. [PDF]. http://www.networkrail.co.uk/wp-content/uploads/2016/11/The-Value-and-Importance-of-rail-Freight-summary-report.pdf. Accessed on 07 February 2019.

Pachl, J., 2004. Railway operations and Control. Vtd Rail Pub.

Piner, D., Condry, D., 2017. International best practices in managing unplanned disruption to suburban rail services. Transp. Res. Procedia 25, 4403–4410.

Rahim, S.K.N.A., Bargiela, A., Qu, R., 2013. Hill Climbing versus genetic algorithm optimization in solving the examination timetabling problem. In: Proceedings of the 2nd International Conference on Operations Research and Enterprise Systems, vol. 1. pp. 43–52.

Schachtebeck, M., Schöbel, A., 2008. IP-Based techniques for delay management with priority decisions. In: ATMOS.

Schachtebeck, M., Schöbel, A., 2010. To wait or not to wait and who goes first? Delay management with priority decisions. Transp. Sci. 44 (3), 307–321.

Schasfoort, B.B.W., 2019. A Decision Support System for real-time track assignment at railway yards. (Master's thesis). University of Twente.

Schneider, J.J., Kirkpatrick, S., 2006. Application of genetic algorithms to TSP. In: Stochastic Optimization. Springer, pp. 415–422.

Schöbel, A., 2007. Integer programming approaches for solving the delay management problem. In: Geraets, F., Kroon, L., Schoebel, A., Wagner, D., Zaroliagis, C. (Eds.), Algorithmic Methods for Railway Optimization. In: Lecture Notes in Computer Science, vol. 4359, Springer, Berlin, Heidelberg, pp. 145–170.

Schöbel, A., 2009. Capacity constraints in delay management. Publ. Transp, 1 (2), 135–154.

Timmermans, P., 2018. Interview with patrick timmermans from prorail, 12 october 2018.

Törnquist, J., 2010. Greedy algorithm for railway traffic re-scheduling during disturbances: A Swedish case. IET Intell. Transp. Syst. 4 (4), 75–386.

Törnquist, J., 2012. Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances. Transp. Res. C 20 (1), 62–78.

Törnquist, J., Persson, J.A., 2005. Train traffic deviation handling using tabu search and simulated annealing, in: Proceedings of the 38th Annual Hawaii International Conference on System Science. pp. 73a–73a.

U.S. Department of Transportation, 1999. Understanding how train dispatchers manage and control trains: a cognitive task analysis of a distributed team planning task. [PDF]. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.876.1554&rep=rep1&type=pdf. Accessed on 14 May 2019.

U.S. Railroad, 2008. A review of capacity and performance data. Accessed on 07 January 2019. [online].

Winter, T., Zimmermann, U., 2000. Real-time dispatch of trams in storage yards. Ann. Oper. Res 96, 287–315.

Zhan, S., Kroon, L.G., Veelenturf, L.P., Wagenaar, J.C., 2015. Real-time high-speed train rescheduling in case of a complete blockage. Transp. Res. B 78, 182–201.

Update

Journal of Rail Transport Planning & Management
Volume 17, Issue , March 2021, Page

DOI: https://doi.org/10.1016/j.jrtpm.2021.100237

Journal of Rail Transport Planning & Management 17 (2021) 100237

Contents lists available at ScienceDirect

Journal of Rail Transport Planning & Management

journal homepage: http://www.elsevier.com/locate/jrtpm

Erratum regarding previously published articles

The Declaration of Competing Interest statements were not included in the published version of articles that appeared in Volume 2 of Journal of Rail Transport Planning & Management. For the below articles, the authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Towards a conflict prevention strategy applicable for real-time railway traffic management [J. Rail Transport. Plann. Manag., 2019; 11(C)] The authors were contacted after publication to request a Declaration of Interest statement.

Open access passenger rail competition in Slovakia – Experience from the Bratislava–Košice line [J. Rail Transport. Plann. Manag., 2019; 12(C)] The authors were contacted after publication to request a Declaration of Interest statement.

A concurrent approach to the periodic event scheduling problem [J. Rail Transport. Plann. Manag., 2019; 15(C)] The authors were contacted after publication to request a Declaration of Interest statement.

Long-distance rail prices in a competitive market. Evidence from head-on competition in Italy [J. Rail Transport. Plann. Manag., 2019; 12(C)] The authors were contacted after publication to request a Declaration of Interest statement.

A dynamic model for real-time track assignment at railway yards [J. Rail Transport. Plann. Manag., 2020; 14(C)] The authors were contacted after publication to request a Declaration of Interest statement.

Competition for and in the passenger rail market: Comparing open access versus franchised train operators' costs and reliability in Britain [J. Rail Transport. Plann. Manag., 2020; 12(C)] The authors were contacted after publication to request a Declaration of Interest statement.

Extending UIC 406-based capacity analysis–New approaches for railway nodes and network effects [J. Rail Transport. Plann. Manag., 2020; 15(C)] The authors were contacted after publication to request a Declaration of Interest statement.

Dynamic assignment model of trains and users on a congested urban-rail line [J. Rail Transport. Plann. Manag., 2020; 14(C)] The authors were contacted after publication to request a Declaration of Interest statement.

Quantifying the effects of running time variability on the capacity of rail corridors [J. Rail Transport. Plann. Manag., 2020; 15(C)] The authors were contacted after publication to request a Declaration of Interest statement.

Optimal running time supplement distribution in train schedules for energy-efficient train control [J. Rail Transport. Plann. Manag., 2020; 14(C)] The authors were contacted after publication to request a Declaration of Interest statement.

A multi-state train-following model for the analysis of virtual coupling railway operations [J. Rail Transport. Plann. Manag., 2020; 15(C)] The authors were contacted after publication to request a Declaration of Interest statement.

Enhanced data reconciliation of freight rail dispatch data [J. Rail Transport. Plann. Manag., 2020; 14(C)] The authors were contacted after publication to request a Declaration of Interest statement.

Maintenance scheduling within rolling stock planning in railway operations under uncertain maintenance durations [J. Rail Transport. Plann. Manag., 2020; 14(C)] The authors were contacted after publication to request a Declaration of Interest statement.

Simulated and real train driving in a lineside automatic train protection (ATP) system environment [J. Rail Transport. Plann. Manag., 2020; 16(C)] The authors were contacted after publication to request a Declaration of Interest statement.

Explaining dwell time delays with passenger counts for some commuter trains in Stockholm and Tokyo [J. Rail Transport. Plann.

DOIs of original article: https://doi.org/10.1016/j.jrtpm.2019.100143, https://doi.org/10.1016/j.jrtpm.2020.100189, https://doi.org/10.1016/j.jrtpm.2019.100176, https://doi.org/10.1016/j.jrtpm.2020.100199, https://doi.org/10.1016/j.jrtpm.2020.100177, https://doi.org/10.1016/j.jrtpm.2019.100144, https://doi.org/10.1016/j.jrtpm.2020.100203, https://doi.org/10.1016/j.jrtpm.2020.100195, https://doi.org/10.1016/j.jrtpm.2020.100196, https://doi.org/10.1016/j.jrtpm.2020.100201, https://doi.org/10.1016/j.jrtpm.2019.100175, https://doi.org/10.1016/j.jrtpm.2019.100142, https://doi.org/10.1016/j.jrtpm.2020.100193, https://doi.org/10.1016/j.jrtpm.2020.100205, https://doi.org/10.1016/j.jrtpm.2020.100198, https://doi.org/10.1016/j.jrtpm.2020.100178, https://doi.org/10.1016/j.jrtpm.2019.07.001, https://doi.org/10.1016/j.jrtpm.2020.100180, https://doi.org/10.1016/j.jrtpm.2020.100206.

https://doi.org/10.1016/j.jrtpm.2021.100237

Available online 12 February 2021

Manag., 2020; 14(C)] The authors were contacted after publication to request a Declaration of Interest statement.

Estimating the demand for rail freight transport in Pakistan: A time series analysis [J. Rail Transport. Plann. Manag., 2020; 14(C)] The authors were contacted after publication to request a Declaration of Interest statement.

Dynamic and robust timetable rescheduling for uncertain railway disruptions [J. Rail Transport. Plann. Manag., 2020; 15(C)] The authors were contacted after publication to request a Declaration of Interest statement.

Urban rail transit operation safety evaluation based on an improved CRITIC method and cloud model [J. Rail Transport. Plann. Manag., 2020; 16(c)] The authors were contacted after publication to request a Declaration of Interest statement.

Hybrid integer-coded Wolf Pack Algorithm for multiple-type flatcars loading problem [J. Rail Transport. Plann. Manag., 2020; 16 (c)] The authors were contacted after publication to request a Declaration of Interest statement.