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

Bram B.W. Schasfoort
University of Twente
Center for Transport Studies (CTS)
Department of Civil Engineering
7500 AE Enschede, The Netherlands
Email: b.schasfoort@duravermeer.nl

Konstantinos Gkiotsalitis
University of Twente
Center for Transport Studies (CTS)
Department of Civil Engineering
7500 AE Enschede, The Netherlands
Email: k.gkiotsalitis@utwente.nl

Oskar A.L. Eikenbroek
University of Twente
Center for Transport Studies (CTS)
Department of Civil Engineering
7500 AE Enschede, The Netherlands
Email: k.gkiotsalitis@utwente.nl

Eric C. van Berkum
University of Twente
Center for Transport Studies (CTS)
Department of Civil Engineering
7500 AE Enschede, The Netherlands
Email: e.c.vanberkum@utwente.nl

99th Annual Meeting of the Transportation Research Board
Paper no. 20-00661

Washington D.C.
January 2020
ABSTRACT
In this paper, we study the real-time train assignment problem (RT-TAP) that arises from the high percentage of unreliable arrival times of freight trains and the large quantity of 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 minutes compared to the solutions of the FSFS heuristic.

Keywords: track assignment; real-time control; rail operations; minimizing delays; rescheduling.
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 (1). As discussed by Givoni and Banister (2), after long and short haul maritime transport, rail freight transport is the mode with the least CO\textsubscript{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 (3). 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, Boysen et al. (4, 5). 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 (6). 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 (7), there is a possibility that unplanned events result in disturbances and/or disruptions. Now, when a delayed freight train arrives at a railway yard, the track needs to be assigned manually by the TD D’Ariano, Josyula et al. (8, 9). 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 (10).

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 hours delayed, and 4% arrived after 24 hours or their original schedule European Commission (11). 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 minutes later than originally scheduled, whereas 26% and 15% of these trains arrive more than one or two hours late, respectively (see Table 1).

| 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         |
| number of trains                                              |               |               |             |              |
| 16                                                           | 8             | 25           | 85          | 40           |
| percentage (%)                                               | 6%            | 3%           | 10%         | 33%          |
| totals:                                                       |               |               |             |              |
| 19% too early                                                 | 33%          | 19% too late | 48% too late |

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 (12). As substantiated by Briggs and Beck (13), 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. 90% of the executed trains are known when making the original planning, which is done 3 days before execution, and 10% of the trains make a last-minute request to park at the yard.

|                      | Year plan | Original plan | Scheduled plan | Executed |
|----------------------|-----------|---------------|----------------|----------|
| number of trains     | 389       | 515           | 573            | 573      |
| percentage (%)       | 66%       | 90%           | 100%           | 100%     |

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 (10). 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.

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.

Train rescheduling
In track assignment problems, most works consider sorting and scheduling problems at marshaling or shunting yards Hansmann and Zimmermann (14). Shunting can be described as the process where the train composition changes on successive train services Boysen et al., Haahr and Lusby, Gatto et al. (5, 15, 16). Jaehn et al., Jaehn et al. (17, 18) 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. (19) developed an integer programming model for scheduling shunting tasks as well as allocating arrival yard tracks and classification bowl tracks. Haahr et al. (20) 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. (21) 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.

Minimizing delays is a multidisciplinary problem that is widely discussed in many different fields Schöbel, Schachtebeck and Schöbel, Schachtebeck and Schöbel (22, 23, 24). A survey on problem models and solution approaches with regards to the rescheduling in railway networks is developed by Fang (25). Several works such as Dollevoet et al. (26) and Schöbel (27) discuss minimizing the total (weighted) delays of trains, and use fast-to-compute heuristics to solve such problems Dollevoet and Huisman (28).

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 Boysen et al., Briggs and Beck, Gilg et al. (5, 13, 21).

**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 (29) 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. (30). An overview of recovery models and algorithms for real-time railway rescheduling was presented by Cacchiani et al. (31).

As discussed by Dollevoet et al. (32), 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. (33). Zhan et al. (33) developed a decision support system (DSS) which found solutions for real-time rescheduling within 10 minutes of computational time. Other models developed by Fischettiia and Monaci (34) and Corman et al. (35) found practical solutions for real-time train rescheduling within seconds. Winter and Zimmermann (36) 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 Fischettiia and Monaci (34) and Corman et al. (35) tried to find practical solutions within seconds, Zhan et al. (33) accepted finding solutions within 10 minutes. 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.

**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. (21) 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. (21) and Winter and Zimmermann (36) are the closest prior arts to our work. However, the model of Gilg et al. (21) 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 (36) 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. (21) and Winter and Zimmermann (36) 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.

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.

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.

We formulate an integer program (IP), well-suited for scheduling and routing problems. We consider a destination yard with a set of tracks \( K = \{1, 2, \ldots, m\} \), and a set of trains \( N = \{1, \ldots, n\} \). 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 Cordeau et al. (37). We introduce graph \( G = (V, E) \) with \( V = \{0, 1, \ldots, n, n+1\} \) 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 \( x^k_{ij} \in \{0, 1\} \) denotes whether (dummy) train \( i \) is scheduled before (dummy) train \( j \) on track \( k \). Variable \( y^k_i \in N, k \in K \), is the time that service of train \( i \) is started on track \( k \). For convenience, we introduce (auxiliary) variable \( z^k_i \) as the finishing time of train \( i \) on track \( k \). \( d_i \) denotes the departure time of train \( i \). Before proceeding to our mathematical program, we summarize the nomenclature.
Sets
\( K = \{1, 2, \ldots, m\} \) set of tracks;
\( N = \{1, 2, \ldots, n\} \) 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 \( k \)

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 \( k \)
\( d_i \) actual departure time of train \( i \)

\begin{table}[h]
\centering
\caption{Nomenclature}
\begin{tabular}{ll}
\hline
\textbf{Parameter} & \textbf{Description} \\
\hline
\( l_i \) & 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 \( k \) \\
\hline
\end{tabular}
\end{table}

The optimization problem RT-TAP can be formulated as follows:

\[
\min \sum_{i \in N} w_i(d_i - \tilde{d}_i) \tag{1}
\]

subject to:

\[
\sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1, \quad i \in N \tag{2}
\]

\[
\sum_{j \in V} x_{0j}^k = 1, \quad k \in K \tag{3}
\]

\[
\sum_{i \in V} x_{i,n+1}^k = 1, \quad k \in K \tag{4}
\]

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

\[
y_i^k \geq a_i, \quad k \in K, i \in V \tag{6}
\]

\[
z_i^k = y_i^k + p_i, \quad k \in K, i \in N \tag{7}
\]

\[
z_j^k \geq z_i^k + H - M(1 - x_{ij}^k), \quad k \in K, (i, j) \in E \tag{8}
\]
\[ d_i \geq z_k, \quad k \in K, i \in N \]  \hspace{1cm} (9)

\[ d_i \geq d_i, \quad i \in N \]  \hspace{1cm} (10)

\[ \sum_{j \in V} x_{ij}^k \leq L_k, \quad k \in K, i \in N \]  \hspace{1cm} (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 \( n + 1 \). (5) is so that for every train a (dummy) train is scheduled before and after it (on the same track). (6) makes that we can only schedule trains after arrival. (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. (37)). Here, \( M \) is a large constant. 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., \( d_i = \max_{k \in K} z_k \). 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. Constraint (11) ensures that trains can only be scheduled on tracks that have sufficient length.

**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. (38)). 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 \( V = \{0,1,\ldots,n,n+1\} \) 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 \( \sum_i d_i = v \).

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.
SOLUTION METHODS

This chapter introduces solution methods used to solve the RT-TAP. Given the complexity of our problem, we introduce a problem-specific genetic algorithm (GA) and a first-scheduled first-search (FSFS) algorithm. Both algorithms do not guarantee the convergence to a globally optimal solution, but can compute an improved solution within a limited amount of time, as it is demonstrated in our numerical experiments.

Problem-specific GA

With the incremental improvements of computer performance over the last decades, we are able to solve larger and more complex problems by using a brute force 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 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. Approximation algorithms, i.e. heuristics, compute solutions based on a solving method that does not evaluate 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, Törnquist (39, 40), hill climbing (HC) Rahim et al. (41), simulated annealing (SA) Törnquist and Persson (42), genetic algorithms (GA) Holland (43) and tabu search (TS) Glover (44). 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 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 (43), 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. (45). 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 \( S \subset \mathcal{P} \) as a (finite) subset of the discrete solution set \( \mathcal{P} \). Each population member \( s \in S \) is an assignment of the \( n \) trains to the \( m \) tracks. Formally,

\[
s = (s_1, s_2, \ldots, s_n),
\]

with

\[
s_i \in \{k \mid l_i \leq L_k\}, \quad i \in N.
\]

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 each \( s \) as formulated in (12) does not correspond to a unique solution, i.e., different assignments may have the same \( s \in S \). 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 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 (relatively fast) using a brute-force algorithm.

Algorithm 1 summarizes our GA-approach (see, e.g., Hurink (46)).

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

Algorithm 1: Genetic Algorithm

We hereby discuss the details of our algorithm. In line 4 of Algorithm 1 we determine the \((\pi - \pi')\), \( 1 \leq \pi' < \pi \), survivors from \( S \). These survivors 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 apply a (random) crossover operation to \( P_i = (r, t) \) with \(|P_i| = 2\) and \( r, t \in S' \). Formally, for each crossover operation we construct a random \( \{0, 1\} \)-vector (see Hurink, Schneider and Kirkpatrick (46, 47))

\[
x = (x_1, x_2, \ldots, x_n)
\]

and we let child \( c = \text{cross}_x(r, t) \) have genes \( c_i = r_i \) if \( x_i = 1 \) and \( c_i = t_i \) if \( x_i = 0 \). In line 8 of Algorithm 1 we mutate the newly-generated members. In fact, each gene \( s_i, i \in N \), is mutated with probability \( p \) to track \( c_i \in \{ k \mid l_i \leq L_k \} \).

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

**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 Schachtebeck and Schöbel, Schöbel (24, 27). 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.

In Line 6 of the Algorithm 2, we solve the following optimization program (for a given train
initialization: Let $N = \{1, 2, 3, \ldots, n\}$ be the set of trains and $K = \{1, 2, 3, \ldots, m\}$ 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 $q_{k^*}^* = \max\{q_{k^*}, a_{i^*}\} + t_{i^*} + H$;
4. end

**Algorithm 2: First-Scheduled First-Served Heuristic**

$i^* \in N$ with corresponding arrival time $a_{i^*}$ and length $l_{i^*}$)

$$(Q_{i^*}) \quad \min_{k \in K} L_k \quad \text{s.t.} \quad L_k \geq l_{i^*} \quad q_k \leq a_{i^*}.$$

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

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

**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.

**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 figure 1.
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).

**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 hours in advance): A scheduled arrival of a train is communicated to the TD 1.5 hours 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.

![FIGURE 1: Schematic overview of Whz process tracks](image-url)
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 35min (on average). Additionally, the original plan of the assignment of trains is presented in Fig.2. Note that in Fig.2 the original schedule considers only 9 out of the 12 available process tracks in Fig.1 because 3 of them are pre-assigned to other activities.

**FIGURE 2**: Original assignment plan of trains in Waalhaven Zuid on Thursday October 1st, 2015.

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.3. 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 hour prior its arrival at the yard is presented in Fig. 4. Note that Fig. 4 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 hour before its actual arrival.

Finally, Fig. 5 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.
FIGURE 5: 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.

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).

The global optimum of our RT-TAP model is found with the use of the CPLEX solver after programming the mathematical model in section 3 in AMPL. To study the optimality gap(s) of the heuristic solution methods, we first use two small-scale scenarios where we consider 13 inbound trains and 3 or 4 tracks available in the yard. Those scenarios are selected because they can be solved with CPLEX, and thus we can compute the optimality gap of the GA and the FSFS. The two small-scale scenarios are tested using the arrival times of trains and their respective lengths, as they were presented in the previous sub-section.

Our third 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 5.2.

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.

Performance on smaller instances
As discussed in section 5.3, the performance is tested in two different experiments. In the first experiment we consider a railway yard with 4 tracks, 13 inbound trains, and the following track lengths: $m_1 = 800$ m, $m_2 = 700$ m, $m_3 = 600$ m, and $m_4 = 500$ m.
We also test the case where the number of available tracks is reduced from 4 to 3, while still considering the same 13 inbound trains. In Table 4 we present the results of those two experiments on small problem instances.

| 4 tracks | Total delay of trains (h:mm:ss) | Running time | 3 tracks | Total delay of trains (h:mm:ss) | Running time |
|----------|--------------------------------|--------------|----------|--------------------------------|--------------|
| CPLEX    | 1:48:36                        | 121 sec      | CPLEX    | 4:41:48                        | 255 sec      |
| FSFS     | 3:29:10                         | < 1 sec      | FSFS     | 6:24:45                        | < 1 sec      |
| GA       | 1:48:36                         | < 1 sec      | GA       | 4:41:48                        | < 1 sec      |

When comparing the GA and the FSFS algorithms with the globally optimal solution of CPLEX, we can conclude that there is no optimality gap between the solution of the GA and CPLEX for smaller instances. However, while the GA converges to the global optimum, FSFS performs rather poorly.

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.6.

![FIGURE 6: Daily performance](image)

From Fig.6, 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 5.
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 sec compared to the reassignments of the FSFS.

**TABLE 5: 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) |
|-----------------|------------------|----------------|---------------------|
| 12:00:00 AM     | 735              | 735            | 0                   |
| 12:30:00 AM     | 735              | 735            | 0                   |
| 1:00:00 AM      | 735              | 735            | 0                   |
| 1:30:00 AM      | 414              | 414            | 0                   |
| 2:00:00 AM      | 414              | 414            | 0                   |
| 2:30:00 AM      | 414              | 414            | 0                   |
| 3:00:00 AM      | 414              | 414            | 0                   |
| 3:30:00 AM      | 414              | 414            | 0                   |
| 4:00:00 AM      | 3830             | 3358           | 472                 |
| 4:30:00 AM      | 3830             | 3358           | 472                 |
| 5:00:00 AM      | 3830             | 3358           | 472                 |
| 5:30:00 AM      | 0                | 0              | 0                   |
| 6:00:00 AM      | 0                | 0              | 0                   |
| 6:30:00 AM      | 0                | 0              | 0                   |
| 7:00:00 AM      | 0                | 0              | 0                   |
| 7:30:00 AM      | 0                | 0              | 0                   |
| 8:00:00 AM      | 0                | 0              | 0                   |
| 8:30:00 AM      | 0                | 0              | 0                   |
| 9:00:00 AM      | 0                | 0              | 0                   |
| 9:30:00 AM      | 0                | 0              | 0                   |
| 10:00:00 AM     | 0                | 0              | 0                   |
| 10:30:00 AM     | 0                | 0              | 0                   |
| 11:00:00 AM     | 0                | 0              | 0                   |
| 11:30:00 AM     | 0                | 0              | 0                   |
| 12:00:00 AM     | 0                | 0              | 0                   |

**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 \((1|r_i]\sum C_i)\)-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.

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 minutes and 42 seconds (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).

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

AUTHOR CONTRIBUTION STATEMENT
The authors confirm contribution to the paper as follows: study conception and design: B.B.W. Schasfoort, K. Gkiotsalitis, E.C. van Berkum; data collection: B.B.W. Schasfoort; analysis and interpretation of results: B.B.W. Schasfoort, K. Gkiotsalitis, O.A.L. Eikenbroek, E.C. van Berkum; draft manuscript preparation: B.B.W. Schasfoort, K. Gkiotsalitis, O.A.L. Eikenbroek. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES
[1] Network Rail, 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, 2016, accessed on 07 February 2019.
[2] Givoni, M. and D. Banister, Moving Towards Low Carbon Mobility. Edward Elgar, 2013.
[3] U.S. Railroad, A Review of Capacity and Performance Data. [online], 2008, accessed on 07 January 2019.
[4] Narayanaswami, S. and N. Rangaraj, Scheduling and Rescheduling of Railway Operations: A Review and Expository Analysis. Technology Operation Management, Vol. vol 2, No. 2, 2011, pp. 102–122.
[5] Boysen, N., M. Fiedner, F. Jaehn, and E. Pesch, Shunting yard operations: Theoretical aspects and applications. European Journal of Operational Research, Vol. vol 220, No. 1, 2012, pp. 1–14.
[6] Pachl, J., Railway operations and Control. Vtd Rail Pub, 2004.
[7] Piner, D. and D. Condry, International best practices in managing unplanned disruption to suburban rail services. Transportation Research Procedia, Vol. vol 25, 2017, pp. 4403–4410.
[8] D’Ariano, A., Improving Real-Time Train Dispatching: Models, Algorithms and Applications. [PhD Thesis] https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=2ahUKEwjhr4e-3JviAhUSDuwKHTmBHlQFjABegQIChAC&url=https%3A%2F%2Frepository.tudelft.nl%2Fislandora%2Fobject%2Fuuid%3A178b886e-d6c8-4d39-be5d-03d9fa3a680f%2Fobject%2Fu7%3Aa178b%2Fdownload&usg=AOvVaw3PKpXVxJTTtJ-NtvKMwGkWx, 2015, accessed on 14 May 2019.
[9] Josyula, S. P., J. T. Krasemann, and L. Lundberg, A parallel algorithm for train rescheduling. Transportation Research Part C, Vol. vol 95, 2018, pp. 545–569.
[10] U.S. Department of Transportation, Understanding how train dispatchers manage and control trains: a cognitive task analysis of a distributed team planning task. [PDF] http://citeseerx.ist.
[11] European Commission, 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, 2014, accessed on 14 May 2019.

[12] Timmermans, P., Interview with Patrick Timmermans from ProRail, 12 October 2018, 2018.

[13] Briggs, N. and C. Beck, Modeling train delays with q-exponential functions. Physica A: Statistical Mechanics and its Applications, Vol. vol 378, No. 2, 2007, pp. 498–504.

[14] Hansmann, R. S. and U. T. Zimmermann, Optimal Sorting of Rolling Stock at Hump Yards, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 189–203, 2008.

[15] Haahr, J. and R. Lusby, Integrating rolling stock scheduling with train unit. European Journal of Operational Research, Vol. vol 259, No. 2, 2017, pp. 452–468.

[16] Gatto, M., J. Maue, M. Mihalak, and P. Widmayer, Shunting for dummies: an introductory algorithmic survey. Springer, 2009.

[17] Jaehn, F., A. Otto, and K. Seifried, Shunting operations at flat yards: retrieving freight railcars from storage tracks. OR Spectrum, Vol. vol 40, No. 2, 2018, pp. 367–393.

[18] Jaehn, F., J. Rieder, and A. Wiehl, Single-stage shunting minimizing weighted departure times. Omega, Vol. vol 52, 2015, pp. 133–141.

[19] Gestrelius, S., M. Aronsson, M. Joborn, and M. Bohlin, Towards a comprehensive model for track allocation and rolltime scheduling at marshalling yards. Journal of Rail Transport Planning Management, Vol. vol 7, 2017, pp. 157–170.

[20] Haahr, J., R. Lusby, and J. Wagenaar, Optimization methods for the Train Unit Shunting Problem. European Journal of Operational Research, Vol. vol 262, No. 3, 2017, pp. 981–995.

[21] Gilg, B., T. Klug, R. Martienssen, J. Paat, T. Schlechte, C. Schulz, S. Seymen, and A. Tesch, Conflict-free railway track assignment at depots. Rail Transport Planning & Management, Vol. vol 8, 2018, pp. 16–28.

[22] Schöbel, A., Integer Programming Approaches for Solving the Delay Management Problem. Springer, Berlin, Heidelberg, 2007, in: Geraets F., Kroon L., Schoebel A., Wagner D., Zaroliagis C.D. (eds) Algorithmic Methods for Railway Optimization. Lecture Notes in Computer Science, vol 4359.

[23] Schachtebeck, M. and A. Schöbel, IP-based Techniques for Delay Management with Priority Decisions. In ATMOS, 2008.

[24] Schachtebeck, M. and A. Schöbel, To wait or not to wait and who goes first? Delay management with priority decisions. Transportation Science, Vol. vol 44, No. 3, 2010, pp. 307–321.

[25] Fang, W., A Survey on Problem Models and Solution Approaches to Rescheduling in Railway Networks. IEEE Transactions on Intelligent Transportation Systems, Vol. vol 16, No. 6, 2015, pp. 2997–3016.

[26] Dollevoet, T., M. Schmidt, and A. Schöbel, Delay Management including Capacities of Stations. In ATMOS, 2011.

[27] Schöbel, A., Capacity constraints in delay management. Public Transport, Vol. vol 1, No. 2, 2009, pp. 135–154.

[28] Dollevoet, T. and D. Huisman, Fast heuristics for delay management with passenger rerouting. Public Transport, Vol. vol 6, No. 1, 2014, pp. 67–84.
[29] Cai, X. and C. Goh, A fast heuristic for the train scheduling problem. *Computers & Operations Research*, Vol. vol 21, No. 5, 1994, pp. 499–510.

[30] Bettinelli, A., A. Santini, and D. Vigo, A real-time conflict solution algorithm for the train rescheduling problem. *Transportation Research part B*, Vol. vol 106, 2017, pp. 237–265.

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

[32] Dollevoet, T., D. Huisman, L. G. Kroon, L. P. Veelenturf, and J. C. Wagenaar, Application of an iterative framework for real-time railway rescheduling. *Computers Operations Research*, Vol. vol 78, 2010, pp. 203–217.

[33] Zhan, S., L. G. Kroon, L. P. Veelenturf, and J. C. Wagenaar, Real-time high-speed train rescheduling in case of a complete blockage. *Transportation Research Part B*, Vol. vol 78, 2015, pp. 182–201.

[34] Fischettia, M. and M. Monaci, Using a general-purpose Mixed-Integer Linear Programming solver for the practical solution of real-time train rescheduling. *European Journal of Operational Research*, Vol. vol 263, 2017, pp. 258–264.

[35] Corman, F., A. D’Ariano, D. Pacciarelli, and M. Pranzo, A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B*, Vol. vol 44, 2010, pp. 175–192.

[36] Winter, T. and U. Zimmermann, Real-time dispatch of trams in storage yards. *Annals of Operational Research*, Vol. vol 96, 2000, pp. 287–315.

[37] Cordeau, J.-F., G. Laporte, M. W. Savelsbergh, and D. Vigo, Vehicle routing. *Handbooks in operations research and management science*, Vol. 14, 2007, pp. 367–428.

[38] Lenstra, J. K., A. R. Kan, and P. Brucker, Complexity of machine scheduling problems. In *Annals of discrete mathematics*, Elsevier, Vol. 1, 1977, pp. 343–362.

[39] Törnquist, J., Greedy algorithm for railway traffic re-scheduling during disturbances: A Swedish case. *IET Intelligent Transport Systems*, Vol. vol 4, No. 4, 2010, pp. 75–386.

[40] Törnquist, J., Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances. *Transportation research. Part C: Emerging technologies*, Vol. vol 20, No. 1, 2012, pp. 62–78.

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

[42] Törnquist, J. and J. A. Persson, Train Traffic Deviation Handling Using Tabu Search and Simulated Annealing. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 2005, pp. 73a–73a.

[43] Holland, J., *Adaptation in natural and artificial systems*. University of Michigan Press, Michigan, 1975.

[44] Glover, F., Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, Vol. vol 13, No. 5, 1986, pp. 533–549.

[45] Gkiotsalitis, K., Z. Wu, and O. Cats, A cost-minimization model for bus fleet allocation featuring the tactical generation of short-turning and interlining options. *Transportation research. Part C: Emerging technologies*, Vol. vol 98, 2019, pp. 14–36.

[46] Hurink, J., *Solving Complex Optimization Problems by Local Search*, 1998.
[47] Schneider, J. J. and S. Kirkpatrick, Application of genetic algorithms to TSP. *Stochastic Optimization*, 2006, pp. 415–422.