

Passenger-centric timetable rescheduling

A user equilibrium approach

Binder, S.; Maknoon, M. Y.; Sharif Azadeh, Sh; Bierlaire, M.

DOI

[10.1016/j.trc.2021.103368](https://doi.org/10.1016/j.trc.2021.103368)

Publication date

2021

Document Version

Final published version

Published in

Transportation Research Part C: Emerging Technologies

Citation (APA)

Binder, S., Maknoon, M. Y., Sharif Azadeh, S., & Bierlaire, M. (2021). Passenger-centric timetable rescheduling: A user equilibrium approach. *Transportation Research Part C: Emerging Technologies*, 132, Article 103368. <https://doi.org/10.1016/j.trc.2021.103368>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.



ELSEVIER

Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

Transportation Research Part C

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

Passenger-centric timetable rescheduling: A user equilibrium approach

S. Binder^a, M.Y. Maknoon^{a,b,*}, Sh. Sharif Azadeh^{a,c}, M. Bierlaire^a^a School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland^b Faculty of Technology, Policy, and Management, Delft University of Technology, Jaffalaan 5, 2628 BX Delft, the Netherlands^c Department of Transport & Planning, Delft University of Technology, Netherlands

ARTICLE INFO

Keywords:

Railway timetabling rescheduling
 Passenger satisfaction
 Passenger assignment
 Pareto frontier
 Heuristic algorithm

ABSTRACT

Unexpected disruptions commonly occur in the railway network, causing delays, and extra cost for operators and inconvenience for passengers by missing their connection and facing overcrowded trains. This paper presents a passenger-centric approach for timetable rescheduling in case of disruption. We study a railway system in which passengers are free to choose their itinerary and compete over limited train capacity. We explicitly model the passengers' decisions using a choice model. We propose a multi-objective algorithmic approach to solve the problem. Service punctuality, operating cost, and passengers' inconvenience are selected as objectives. Computational experiments are performed on the Swiss and Dutch railway networks. The results demonstrate the performance of the algorithm in finding high-quality solutions in a computationally efficient manner.

1. Introduction

Dealing with disruption is an inevitable part of the railway's daily operations. Events such as open-track blockade, train station shutdown, and adverse weather conditions interrupt rail service. As reported in [Riden de Treinen \(2021\)](#), the Dutch railway faced nine disruptions per day between 2011 and 2021. These disruptions last on average 2.5 h. As a result, passengers experience delays, miss the connecting trains, and face overcrowded trains.

During a disruption, the initial timetable becomes infeasible. As a result, the operator plans a disposition timetable to restore the train service while minimizing passengers' inconvenience. As noted by [Batley et al. \(2011\)](#), during the disruption, passengers rarely change their travel mode. Instead, they adapt their itinerary and departure time, see, [Benezech and Coulombel \(2013\)](#) and [Parbo et al. \(2016\)](#). There are two common approaches to incorporate passengers' behavior into designing the disposition timetable. Passengers' behavior is either *implicitly* or *explicitly* considered in the planning process.

In the first approach, passengers' behavior is *implicitly* considered in the disposition timetable aimed to minimize delays and cancellations of trains, see, for example, [Narayanaswami and Rangaraj \(2013\)](#), [Louwerse and Huisman \(2014\)](#), [Veelenturf et al. \(2016a\)](#) and, [Ghaemi et al. \(2018\)](#). However, minimizing the train delays creates conflicts with passengers' connections, ultimately delaying their journey.

* Corresponding author at: Faculty of Technology, Policy, and Management, Delft University of Technology, Jaffalaan 5, 2628 BX Delft, the Netherlands.

E-mail addresses: s.binder@epfl.ch (S. Binder), M.Y.Maknoon@tudelft.nl (M.Y. Maknoon), sharifazadeh@ese.eur.nl (Sh. Sharif Azadeh), michel.Bierlaire@epfl.ch (M. Bierlaire).

<https://doi.org/10.1016/j.trc.2021.103368>

Received 14 February 2021; Received in revised form 2 August 2021; Accepted 26 August 2021

Available online 17 September 2021

0968-090X/© 2021 The Author(s).

Published by Elsevier Ltd.

This is an open access article under the CC BY license

(<http://creativecommons.org/licenses/by/4.0/>).

In the second approach, the passengers' itinerary is *explicitly* considered. The goal is to reschedule the trains in a way that the total passengers' delay is minimized. Here, the passengers' itineraries are either assumed to be static or dynamic. In the static case, the passenger itinerary is the one that is realized according to the initial timetable, see, [Zhu and Goverde \(2019\)](#). In contrast, in the dynamic case, it is assumed that passengers adapt their itinerary to the changes in the timetable. The passengers' inconvenience is expressed as a function of delay, waiting time, in-vehicle time, etc. The disposition timetable aims at minimizing the overall inconvenience of passengers, see [Binder et al. \(2017b\)](#), [Zhu and Goverde \(2020b\)](#) and [Zhan et al. \(2021\)](#). In the above-mentioned approaches, a system optimum solution (minimizing the total inconvenience for all passengers) is provided for the disposition timetable. In practice, a system optimum solution can only be applied to the railway systems using a seat reservation mechanism. However, it has practical limitations for those systems in which passengers are free to choose their itineraries. Further, estimating passengers' inconvenience becomes more complicated if the number of boarding passengers exceeds the available capacity. Therefore, the disposition timetable needs to consider passengers' routes and estimate who can board the train and who cannot.

For railway service, passengers assignment models are well-studied problems to predict travelers' behavior, see, [Binder et al. \(2017a\)](#) and [Zhu and Goverde \(2020a\)](#). These models provide insight about the train occupancy and enable the decision-maker to comprehend the service quality perceived by travelers. At the same time, any change in the disposition timetable influences travelers behavior which complicates the integration of passenger assignment model into designing disposition timetables. This paper presents an algorithmic framework to design the disposition timetable when passengers could freely choose their itinerary, and the train capacity is limited. We consider passengers' inconvenience, operating cost, and deviation from the initial timetable as rescheduling objectives. The disposition timetable is determined by retiming, reordering, canceling, rerouting, and short-turning the initially planned timetable or establishing the emergency trains and buses. We apply our algorithm to the Swiss and Dutch railway networks. The results show that our proposed approach can find high-quality solutions in a computationally efficient manner.

The remainder of this paper is as follows. Section 2 reviews recent contributions to train rescheduling problems. Description of the problem is presented in Section 3. The overview of our resolution approach is described in Section 4. The details of the algorithm is presented in Sections 5–7. In Section 8, we report the computational experiments on our selected case studies. Finally, Section 9 concludes the paper and provides directions for future research.

2. Related literature

Dealing with service interruption is either studied as disturbance or disruption management. Generally, in disruption management, the incidents block railway tracks. The initial timetable becomes infeasible, as a result, a disposition timetable is designed to recover the train service. In this section, we review the relevant research dealing with disruption management. Interested readers are referred to [Cacchiani et al. \(2014\)](#) for a comprehensive review of railway recovery algorithms.

Disposition timetable is determined by modifying the plan of the initial scheduled trains. The commonly used decisions are retiming, reordering, canceling, flexible stop, short-turning and, adding trains. From the operator's perspective, it is preferred to maintain service punctuality while recovering from disruption. This way, the operator can control the negative impact of disruptions. The service punctuality is measured by indicators such as minimizing the average delay, minimizing the maximum delay, minimizing train cancellation, or their combinations. At microscopic level, this problem has been studied by [D'Ariano et al. \(2007\)](#), [Corman et al. \(2011\)](#), [Corman et al. \(2012\)](#), [Törnquist and Persson \(2007\)](#) [Meng and Zhou \(2014\)](#) and [Pellegrini et al. \(2014\)](#). Other studies, tackle the problem at macroscopic level, see, [Narayanaswami and Rangaraj \(2013\)](#), [Louwerse and Huisman \(2014\)](#), [Veelenturf et al. \(2016a\)](#), [Zhan et al. \(2015\)](#), [Zhan et al. \(2016\)](#) and [Ghaemi et al. \(2018\)](#). Algorithmic resolution methods have been presented in [Brucker et al. \(2002\)](#), [Albrecht et al. \(2013\)](#), [Zhu and Goverde \(2021\)](#) and [Corman et al. \(2014\)](#).

The disposition timetable designed from the operator's perspective neglects passengers' traveling route. Therefore, the new timetable may break passengers' transfer connections and ultimately delay passengers' journeys. To overcome this challenge, the delay management problem, introduced by [Schöbel \(2001\)](#), considers passengers itinerary when modifying the trains schedule. The goal is to minimize the total delay of passengers. This is achieved by synchronizing trains schedule at stations. In other words, keeping the departing train to allow boarding passengers on delayed arriving one, see [Schöbel \(2009\)](#), [Schachtebeck and Schöbel \(2010\)](#), [Dollevoet et al. \(2012\)](#), [Corman et al. \(2016\)](#), [Espinosa-Aranda and García-Ródenas \(2013\)](#), [Kanai et al. \(2011\)](#) and [Corman \(2020\)](#).

Disruption causes variation in travel demand and passengers' route choice compared to the operation of a regular day. Rather than reducing the passengers' delay, several studies quantify overall passengers' experience for designing the disposition timetable. [Zhu and Goverde \(2019\)](#) proposed a two-step approach where first, they estimate the passengers' route choice on a regular day using a schedule-based passenger assignment model. Then, the estimated routes are used in the rescheduling model to minimize passengers inconvenience (see also [Cadarsó et al. \(2013\)](#) and [Sato et al. \(2013\)](#)). The model is extended in [Binder et al. \(2017b\)](#), [Zhu and Goverde \(2020b\)](#) and [Zhan et al. \(2021\)](#) in which a system optimum solution is provided to explicitly incorporate passengers' routes. The system optimum solution cannot be applied to the railway systems in which passengers freely choose their travel routes. This is critical for designing the disposition timetable for densely populated areas where the train's capacity plays an important role.

The seminal works of [Niu and Zhou \(2013\)](#) and [Tian and Niu \(2020\)](#) studied a saturated railway corridor in which passengers demand is time-dependent. The goal is to schedule trains to satisfy the passengers demand. A closely related problem is also studied in [Chen et al. \(2019\)](#) and [Almodóvar and García-Ródenas \(2013\)](#). For a given disposition timetable, the rolling stock scheduling problem aims to determine the optimal composition of rolling stocks. [Kroon et al. \(2015\)](#) present a mathematical model for rolling stock scheduling and an algorithm for passengers assignments. Due to the difficulty resulted from combining passengers' assignment with rolling stock scheduling model, they propose an iterative approach to minimize the system-related cost (e.g., penalty to modify rolling stock composition) and service-related cost (e.g., the effect of train capacity on the delay of passengers). [Veelenturf \(2014\)](#) extend the

passenger assignment model proposed by Kroon et al. (2015) to reschedule rolling stock by changing trains stopping pattern. Finally, van der Hurk et al. (2018) present an algorithm to minimize the passenger inconvenience by simultaneously deciding about the passengers advised routes (i.e. which routes should be suggested to a passenger) and rolling stock schedule.

In summary, existing literature on the passenger-centric approach focuses on providing a system optimum solution that has limited applicability when the passengers are free to choose their itinerary. To overcome this challenge, in this paper, we propose an algorithmic framework to design a disposition timetable taking into account the train capacity. We assume that passengers choose their route freely and look to find a user equilibrium solution. We present a multi-objective optimization approach considering passengers' inconvenience, operator's cost, and service punctuality. We introduce a tailored algorithmic approach to explore non-dominated solutions for all objectives in a computationally efficient manner.

3. Problem description

We consider a railway network with several train lines in which one or multiple tracks become unavailable, resulting from the disruption. We assume that the duration of disruption is known, and we are interested in designing a disposition timetable over the recovery period. During the disruption period, the system performance (e.g., track capacity) is reduced. Once we introduce the disposition timetable, the system performance will improve. After the end of disruption, there is a transition to the normal operating condition. We assume that the disruption period happens within the recovery period. If the train company knows the disruption starting time, the recovery period can begin before the disruption. Otherwise, it starts at the same time as the disruption happens and ends after it is over. This way, the disposition timetable includes the transition to normal operating conditions. All individual trains are scheduled to run along the train line, but trains may alter their course during the recovery period.

In this research, the travel demand between each origin–destination pairs is known. Passengers are characterized by the origin–destination stations and departure time. The passengers know the schedule of trains in the disposition timetable and freely choose their itinerary. We suppose that passengers are utility maximizer agents and choose an itinerary that provides them the highest satisfaction level. However, they have no information about the crowdedness of the trains. In Section 6, we describe our approach to model passengers behavior.

We define a *block train* as a combination of rolling stocks with known capacity. The capacity of a *block train* is measured as the maximum number of onboard passengers and is fixed during the planning horizon. Coupling and decoupling of trains are not considered. The disposition timetable is designed by either modifying the schedule of the block trains operating in the original timetable (i.e., retiming, canceling, rerouting, turn-around) or adding additional *block trains* and buses. Moreover, we assume that the composition of the additional *block trains* (in terms of capacity) and their associated shunting yards are known to the planner.

We study the problem at the macroscopic level. Meanwhile, we consider some headway-related constraints to increase the chance of finding a feasible solution at the microscopic level. The performance of our proposed approach to handle disruption is investigated from the interaction between the disposition timetable, train capacity, and passengers behavior. We quantify these interactions through three objectives: (1) minimizing the operating cost, (2) minimizing the passengers inconvenience, and (3) minimizing the deviation from the original timetable. These three objectives also show the preferences of the key actors of the railway service, i.e., the train operating company, passengers, and infrastructure manager.

3.1. Mathematical model

For a given disruption and a recovery period, let $G \in \mathcal{G}$ be the set of all possible disposition timetables. We define $\omega(G)$ as a function describing how the passengers choose their itinerary if timetable G is selected. The output of this function shows the passenger flow which is noted by \mathcal{J} . The disposition timetable can be presented as:

$$\min (z_{\mathcal{D}}, z_{\mathcal{C}}, z_{\mathcal{I}}) \quad (1)$$

$$G \in \mathcal{G} \quad (2)$$

$$\mathcal{J} = \omega(G) \quad (3)$$

The model (1)–(3) has three objectives: (1) $z_{\mathcal{D}}$ measures the deviation from the original timetable (i.e., service punctuality), (2) $z_{\mathcal{C}}$ evaluates the operating cost of the disposition timetable, and (3) $z_{\mathcal{I}}$ measures the passengers inconvenience due to disruption. In Section 7.1, we describe the procedure to calculate the value of these objectives.

The optimization model (1)–(3) is hard to solve for three main reasons. First, there is a temporal relation between the decision of passengers and the rescheduled timetable. The proposed timetable constrains the decisions of passengers and vice versa. Second, we assume that passengers are free to choose their itinerary and cannot be controlled by the operator. Any model that seeks to optimize both will result in a system optimum solution, whereas our interest is to find a user equilibrium solution. Third, due to the limited train capacity, it is not easy to decide who can board the train and who cannot. As a result, in this paper, we focus on developing an algorithmic framework to capture the dynamism mentioned above. We aim at providing a flexible model to handle various decisions to modify the timetable. In the next section, we describe the general framework of our proposed algorithm.

4. Algorithmic overview

As mentioned in Section 3.1, model (1)–(3) is a multi-objective optimization problem. To overcome the challenge of this capacitated mathematical model, we present an algorithmic framework. The main idea is to combine a heuristic (based on the adaptive large neighborhood search) with a passenger assignment model. The general steps of our approach are presented in Fig. 1 and detailed in Algorithm 1.

The algorithm uses the original timetable (noted by G^0) as an input and provides a set of non-dominated timetables showing the trade-off between the three stated objectives (note by \mathcal{A}). We use the principles of simulated annealing as stopping criteria. Let T_i^{start} and T_i^{end} be the starting and ending temperature for each objective, $i \in \{z_P, z_O, z_D\}$. The iterative process continues till the temperature of all the objectives reach its ending value.

Inside the loop, the algorithm follows an iterative approach. Step 3 applies different operators to design a new feasible timetable (Section 5). The feasible timetable is then used in a passenger assignment model in step 4 to estimate passengers flow (Section 6). The timetable's performance is evaluated in step 5 (Section 7.1) and archived in step 6. In steps 7 to 9, the algorithm periodically updates the temperature and weights of the operators. In Section 7, we will describe the details of steps 6 to 9.

Algorithm 1. General steps of the hyper-heuristic

Input: Initial timetable G^0
 Recovery period H
 Initial (final) temperatures T_i^{start} (T_i^{end}), $i \in \{z_P, z_O, z_D\}$
Output: Collection of feasible non-dominated timetables \mathcal{A}

```

1  $T_i \leftarrow T_i^{start}$ ,  $i \in \{z_P, z_O, z_D\}$ ,  $\mathcal{A} \leftarrow \emptyset$ ,  $G, G' \leftarrow G^0$ 
2 while  $T_i > T_i^{end}$ ,  $i \in \{z_P, z_O, z_D\}$  do
3    $G' \leftarrow$  Adjusting Timetable,  $G$  – (Section 5)
4    $\mathcal{I} \leftarrow$  Assigning Passengers,  $\omega(G')$  – (Section 6)
5   Evaluate  $z_P, z_O, z_D$  – (Section 7.1)
6    $\mathcal{A} \leftarrow$  Archiving  $X(G', \mathcal{I})$  – (Section 7.2)
7   Periodically, update  $T_i$  – (Section 7.3, Eq. (6))
8   Periodically, update weights of operators (Section 7.4)
9   Periodically, select  $X$  randomly in  $\mathcal{A}$  (Section 7.5)
10 return  $\mathcal{A}$ 

```

4.1. Timetable representation

We present the railway infrastructure with a set of stations, $s \in S$, and a set of tracks, Q . A track $(s, s') \in Q$ is defined as a railway connecting station s to s' directly without passing through any other station. We define two stations s and s' to be neighboring if both (s, s') and (s', s) belongs to Q . Tracks can be used by trains in both directions. In any case, a certain headway has to be respected if two

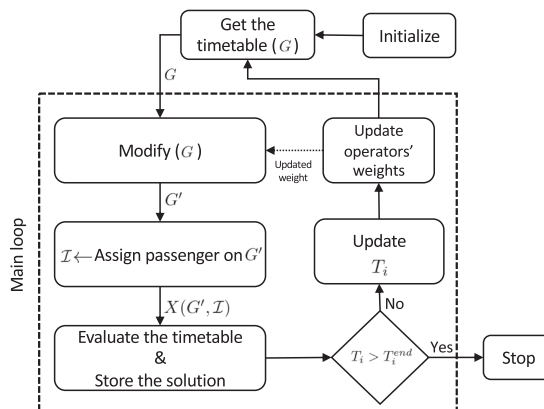


Fig. 1. Algorithm overview.

trains are scheduled on the same track. The running time and travel distance between two stations are symmetric and shown by $t(s, s')$, in minutes and $d(s, s')$, in kilometers, respectively. Trains cannot switch tracks between stations, and overtaking occurs only within stations. Finally, we define a train line $\ell \in L$ as a succession of neighboring stations visited in a given order.

We discretize the recovery period into $n + 1$ time intervals (τ), typically one minute, and present it by the set $H := \{0, \tau, 2\tau, \dots, n\tau\}$. The disposition timetable can then be presented on a time–space graph with $V = S \times H$ nodes. Let $k \in K = \{K_1 \cup K_2\}$ be a set of available *block trains*. Set K_1 shows the trains operating according to the undisturbed timetable. Set K_2 is the *block train* parked in the shunting yards. We call the trains in K_2 emergency trains. The capacity of each block train is noted by q_k . We present the schedule of train $k \in K$ as a path in the time–space graph and denote it by r_k . Set R shows the schedule of all trains in the network. Disposition timetable is presented by $G = (V, R)$. For a given timetable, the passenger assignment model, $\omega(G)$, determines the passengers flow shown by \mathcal{S} . The interaction between the disposition timetable and passengers decision is shown by $X = (G, \mathcal{S})$.

5. Timetable adjustment

This section describes the procedure to modify the exiting timetable (G) to create a disposition timetable (G'). Inspired by the Adaptive Large Neighborhood Search (ALNS) first introduced by Ropke and Pisinger (2006), we define several operators to modify the timetable. Algorithm 2 presents the general steps of our proposed approach. We define two sets of operators (Π and Π_{Inf}) that can be applied on G . If timetable G is feasible (Section 5.1), operators in the set Π are utilized to find a new one. Otherwise, we use several operators (noted by set Π_{Inf}) to restore the feasibility. The operators are selected probabilistically using the roulette wheel mechanism described in Section (7.4).

5.1. Timetable feasibility

We use macroscopic operational constraints to check whether a timetable is feasible or not. A timetable is feasible if it does not have *headway-related* and *disruption-related* infeasibility. The detail of each case is described as follows:

A *headway-related infeasibility* occurs when two trains do not respect each other's minimum headway while they are scheduled to run on the same track. If both trains are running in the same direction, a minimal headway time h needs to be respected. If two trains are scheduled to run in opposite directions on a given track (s, s'), we need to ensure that the track is vacated by the first train before the second one can use it. Again, we consider a minimum headway time h needs to be respected between trains. Hence, the difference between the two trains departure times from station s and s' has to be greater than $h + t(s, s')$.

A *disruption-related infeasibility* occurs when a train is scheduled on a track that is not available due to disruption. More precisely, in case of disruption between station s and s' , from time t_1 to t_2 , a train movement from station s to s' (or vice versa) is said to be infeasible if its departure time or its journey (between s to s') falls in the time interval between t_1 and t_2 . Note that initially, only disruption-related infeasibilities occur since the track is unavailable. Later on, both types of infeasibilities may appear simultaneously and multiple times after applying the operators.

Algorithm 2. Adjusting timetable

```

Input : Timetable  $G$ 
          Set of general operators  $\Pi$ (Section 5.2)
          Set of infeasibility operators  $\Pi_{Inf}$  (Section 5.2)
Output : Feasible timetable  $G'$ 
1 if  $G$  is feasible (Section 5.1) then
2   |  $\pi \leftarrow$  Select operator from  $\Pi$  (Section 5.2)
3   |  $G' \leftarrow$  Apply  $\pi$  to  $G$ 
4 if  $G'$  is not feasible (Section 5.1) then
5   | while  $G'$  is not feasible do
6     |  $\pi_{Inf} \leftarrow$  Select infeasibility operator from  $\Pi_{Inf}$  (Section 5.2)
7     |  $G'' \leftarrow$  Apply  $\pi_{Inf}$  to  $G'$ 
8     | Update score of  $\pi_{Inf}$  (Section 7.4)
9     |  $G' \leftarrow G''$ 
10  | Update weights of  $\pi_{inf} \in \Pi_{Inf}$  (Section 7.4)

```

5.2. Operators

We define an operator as a combination of two building blocks: a *neighborhood structure* that defines how to modify a timetable to generate a neighbor timetable, and a *search strategy* that modifies the current timetable by selecting one of the neighbors. In the

following, we first introduce ten neighborhood structures and then we present the search strategies associated with each of them.

5.2.1. Neighborhood structures

The neighborhood structures introduced in this section define the potential modifications to obtain a neighbor timetable. Table 1 provides a summary of these neighborhood. The detailed description of the neighborhood structure is described below.

Cancellation (CC, PC). Cancellation neighborhood structures select a train $k \in K$ and cancel it, either completely (CC) or partially (PC). In partial cancellation, the train is canceled after a station $s \in S_k$, where $S_k \subset S$ is the ordered set of stations visited by train k .

Delay (CD, PD). Delay neighborhood structures modify a train $k \in K$ by delaying it, either completely (CD) or partially (PD). We assume that the time by which the train is delayed is a multiple of 5 min and cannot exceed 30 min. We introduce the set of potential delay times, $H_D = \{5, 10, 15, 20, 25, 30\}$. In the case of a complete delay, train k is kept behind by $t \in H_D$ minutes on its entire line. For a partial delay after station $s \in S_k$, train k is operated on time up to station s and delayed by $t \in H_D$ minutes after that.

Advance (CA, PA). Advance neighborhood structures modify a train $k \in K$ by scheduling it earlier, either completely (CA) or partially (PA). The following additional conditions need to be respected. i) if k is an original train, no departure time can be scheduled earlier than in the undisturbed timetable. ii) a train cannot depart from a station earlier than its departure time from the previous station, plus the running time between the stations and the minimal dwell time at the station. For train k , we define $H_A^k \subseteq H_D$ as the set of possible advance time respecting the above-mentioned constraints. In the case of a complete advance, train k is advanced by $t \in H_A^k$ minutes on its entire line. For a partial advance after station $s \in S_k$, train k is not modified up to station s , and advanced by $t \in H_A^k$ minutes after that.

Rerouting (RR). The rerouting neighborhood structure (RR) modifies a train $k \in K$ by rerouting it between neighboring stations $s, s' \in S_k$. The set of all possible reroutings between stations s and s' is generated in advance, and the one with the minimum deviation time is selected. Let $\{s_j\}_{j=1}^N, N \geq 1$, be the sequence of stations visited in this rerouting between stations s and s' . Train k is then modified in the following way: up to station s , there is no modification to its schedule; between s and s' , the train is rerouted along with stations $\{s_j\}_{j=1}^N$; after s' , the train runs on its original line but is delayed because of the rerouting. The departure times at stations after s are updated according to running and minimal dwell times.

Turn-around (TA). The turn-around neighborhood structure (TA) modifies a train $k \in K$ by turning it around after station $s \in S_k$. Train k is modified in the following way: up to station s , there is no modification to its schedule. Then, a turn-around time of 10 min is considered, after which it departs in the opposite direction, back to the station where it started its journey. The departure times at stations after s are updated according to running and minimal dwell times.

Emergency Train (IT). The neighborhood structure inserting an emergency train (IT) schedules a new train along line $\ell \in L$. An additional train can be scheduled every 15 min from the first station of the line.

Emergency Bus - (IB). The neighborhood structure inserting an emergency bus (IB) schedules a new bus between two neighboring stations $s, s' \in S$. A bus can be scheduled to depart every 5 min. For simplicity, we assume that the bus travel time and cost are doubled compared to operating a train between the stations.

5.2.2. Search strategies

The choice of parameters (k, s, s', t, ℓ) of the neighborhood structures described above determines which neighboring timetable G' is constructed from timetable G . Four search strategies are used to choose the combination of parameters from this list: local search, random selection, disruption mitigation, and feasibility restoration.

Local search. A local search algorithm is run to select a combination of parameters that minimizes locally one of the objectives $(z_{\mathcal{P}}, z_{\mathcal{C}}$ or $z_{\mathcal{F}}$). This search strategy is applied in association with the following neighborhood structures: CC, PC, CD, PD, CA, PA, or IT. Ideally, the set of all possible neighborhood timetables $N(G)$ should be constructed by applying one of the above neighborhood structures with all possible parameter combinations. However, the construction of the complete set $N(G)$ becomes computationally demanding. Therefore, we impose an upper bound ν_{\max} on the number of evaluated neighbors. For a given neighborhood structure, the local search considers successively the feasible parameter combinations introduced in Table 1 and adds the neighbor timetable to $N(G)$.

Table 1
List of neighborhood structures.

Name	Label	Parameters	Description
Complete Cancel	CC	k	Select train $k \in K$ and cancel it completely
Partial Cancel	PC	k, s	Select train $k \in K$ and cancel it after its arrival in station $s \in S_k$
Complete Delay	CD	k, t	Select train $k \in K$ and delay it by $t \in H_D$ minutes on its complete line
Partial Delay	PD	k, t, s	Select train $k \in K$ and delay it by $t \in H_D$ minutes after station $s \in S_k$
Complete Advance	CA	k, t	Select train $k \in K$ and advance it by $t \in H_A^k$ minutes on its complete line
Partial Advance	PA	k, t, s	Select train $k \in K$ and advance it by $t \in H_A^k$ minutes after station $s \in S_k$
Insert Train	IT	ℓ, t	Insert an emergency train along train line $\ell \in L$, starting its trip at time $t \in H_t^{\ell}$
Reroute	RR	k, s, s'	Select train $k \in K$ and reroute it between neighboring stations $s, s' \in S_k$
Turn Around	TA	k, s	Select train $k \in K$ and turn it around at station $s \in S_k$
Insert Bus	IB	s, s', t	Insert a bus between neighboring stations $s, s' \in S$, starting at time $t \in H_B$

Once $|N(G)| = \nu_{\max}$, the neighbor timetable $G' \in N(G)$ with the lowest value of the corresponding objective function is selected. Here, we have to note that the operating cost is expressed as the kilometer cost of running the train. The delay and advance neighborhood structure do not affect the timetable's operational cost. Therefore, the local search minimizing $z_{\mathcal{D}}$ is not defined for CD, PD, CA, and PA. In Section 7.1 we explain how the values of the objectives are calculated.

Random selection. This search strategy is applied in association with all neighborhood structures, except RR and TA. The main reason behind this strategy is to diversify the search space. For the selected neighborhood structures, the associated parameters are randomly selected from their corresponding sets.

Disruption mitigation. This search strategy is applied exclusively in association with the bus insertion neighborhood structure. Its purpose is to schedule a bus that "bridges the gap" between the two sides of the disruption. For a disruption on track (s_1, s_2) , the neighborhood structure IB is used with $s = s_1$ and $s' = s_2$. Two operators are defined: the departure time t is either drawn uniformly from H_B , or a local search is applied to determine the value $t \in H_B$. Again, an upper bound ν_{\max} is imposed on the number of evaluated timetables in the local search.

Feasibility restoration. This search strategy is applied in association with all neighborhood structures, except IT and IB. Starting from an infeasible timetable G' (according to the definition of Section 5.1), its purpose is to generate a feasible neighbor timetable G'' . The search strategy uses the first infeasibility of the timetable to determine the appropriate combination of parameters for a given neighborhood structure. If the infeasibility is *headway-related*, let $k_1, k_2 \in K$ be the two involved trains and $(s_1, s_2) \in Q$ be the track where the minimal headway is not respected. In this case, the parameters of the neighborhood structure are set (when applicable) to $k = k_1, s = s_1, s' = s_2$ and $t = h$ (the minimal headway time). If the infeasibility is *disruption-related*, let $k_1 \in K$ be the train that is scheduled on the disrupted track $(s_1, s_2) \in Q$. In that case, the parameters of the neighborhood structure are set (when applicable) to $k = k_1, s = s_1$ and $s' = s_2$. The delay time t is selected (in H_D or $H_A^{k_1}$) so as to train k_1 passes track (s_1, s_2) just before the start or after the end of the disruption. If no such time exists, t is set to the maximal delay or advance time. Note that one application of a feasibility restoration operator does not necessarily result in a feasible neighbor timetable. Therefore, a loop is introduced in Algorithm 2 (lines 5–10) to ensure that the final neighbor timetable G'' is feasible before its potential inclusion in the archive.

Summary. Based on the above descriptions, Table 2 summarizes the operators that are used in our framework. Each row corresponds to a neighborhood structure, and each column indicates the search strategy that determines the parameters of the neighborhood structure. A total of 35 operators are thus defined. We classify the operators into two sets: *general operators* (Π) and *infeasibility operators* (Π_{Inf}). The former tries to improve the current solution, whereas the latter attempts to make the current solution feasible. The set of general operators consist of $\Pi = \{\Pi_C \cup \Pi_D \cup \Pi_A \cup \Pi_I, \Pi_B\}$ in which $\Pi_C = \{CC_{\mathcal{D}}, CC_{\mathcal{D}}, CC_{\mathcal{R}}, CC_{\mathcal{R}}, PC_{\mathcal{D}}, PC_{\mathcal{D}}, PC_{\mathcal{R}}, PC_{\mathcal{R}}\}$ is "cancel operators", $\Pi_D = \{CD_{\mathcal{D}}, CD_{\mathcal{D}}, CD_{\mathcal{R}}, PD_{\mathcal{D}}, PD_{\mathcal{D}}, PD_{\mathcal{R}}\}$ is "delay operators", $\Pi_A = \{CA_{\mathcal{D}}, CA_{\mathcal{D}}, CA_{\mathcal{R}}, PA_{\mathcal{D}}, PA_{\mathcal{D}}, PA_{\mathcal{R}}\}$ is "advance operators", $\Pi_I = \{IT_{\mathcal{D}}, IT_{\mathcal{D}}, IT_{\mathcal{R}}, IT_{\mathcal{R}}\}$ is the "train insertion operators" and, $\Pi_B = \{IB_{\mathcal{D}}, IB_{\text{Dis}, \mathcal{D}}, IB_{\text{Dis}, \mathcal{R}}\}$ is the "bus insertion operators". The set of infeasibility operators is given by $\Pi_{\text{Inf}} = \{CC_{\text{Inf}}, PC_{\text{Inf}}, CD_{\text{Inf}}, PD_{\text{Inf}}, CA_{\text{Inf}}, PA_{\text{Inf}}, RR_{\text{Inf}}, TA_{\text{Inf}}\}$.

Here, each operator aims at resolving the infeasibility issues by applying a specific rule. The selection of these operators depends on their performance in resolving the infeasibility. Therefore, for each timetable a unique combination of infeasibility operators are applied. We will discuss operators' selection in more details in Section 7.4.

6. Passengers assignment

For each constructed disposition timetable, a new evaluation of the passengers flow is required. When the trains are overcrowded, it is challenging to estimate who can board the train. This section presents an efficient stochastic scheduled-based passengers assignment algorithm that explicitly considers the capacity of *block trains*. We extend and adapt the passenger assignment model proposed by Binder et al. (2017a). The algorithm's main idea is to define an exogenous priority list showing the order in which passengers are assigned. With this approach, the algorithm can accurately estimate the passengers' flow in a computationally efficient manner.

The scheduled-based passenger assignment algorithm takes the following information as input: a train timetable (G), passengers'

Table 2
List of operators.

	Local search			Search strategy	Disruption mitigation	Feasibility restoration
	$z_{\mathcal{D}}$	$z_{\mathcal{D}}$	$z_{\mathcal{D}}$	Random selection		
CC	$CC_{\mathcal{D}}$	$CC_{\mathcal{D}}$	$CC_{\mathcal{D}}$	$CC_{\mathcal{R}}$	-	CC_{Inf}
PC	$PC_{\mathcal{D}}$	$PC_{\mathcal{D}}$	$PC_{\mathcal{D}}$	$PC_{\mathcal{R}}$	-	PC_{Inf}
CD	$CD_{\mathcal{D}}$	-	$CD_{\mathcal{D}}$	$CD_{\mathcal{R}}$	-	CD_{Inf}
PD	$PD_{\mathcal{D}}$	-	$PD_{\mathcal{D}}$	$PD_{\mathcal{R}}$	-	PD_{Inf}
CA	$CA_{\mathcal{D}}$	-	$CA_{\mathcal{D}}$	$CA_{\mathcal{R}}$	-	CA_{Inf}
PA	$PA_{\mathcal{D}}$	-	$PA_{\mathcal{D}}$	$PA_{\mathcal{R}}$	-	PA_{Inf}
IT	$IT_{\mathcal{D}}$	$IT_{\mathcal{D}}$	$IT_{\mathcal{D}}$	$IT_{\mathcal{R}}$	-	-
RR	-	-	-	-	-	RR_{Inf}
TA	-	-	-	-	-	TA_{Inf}
IB	-	-	-	$IB_{\mathcal{R}}$	$IB_{\text{Dis}, \mathcal{D}}, IB_{\text{Dis}, \mathcal{R}}$	-

list (P), and a list defining assignment priorities among the passengers (\mathcal{L}). In the following sub-sections, we present the key elements of the passengers assignment model.

6.1. Passenger's decision

We assume that demand is known in the form of an origin–destination (OD) matrix per time interval (τ). Let $p \in P$ be a set of passengers entering the system during the disruption. We show each passenger by a tuple (o_p, d_p, t_p) . Here, $o_p, d_p \in S$ show the origin and destination stations and $t_p \in H$ is the desired departure time of passenger p .

Let $j \in \mathcal{J}^p$ be the set of all possible itineraries for passenger p in the disposition timetable. We associate a utility value with each itinerary and call it $U(j)$. It is calculated by $\mathbf{Z}_j \cdot \beta$ in which \mathbf{Z}_j is a vector of attributes and β is a weight vector. The following specifications are considered as the values of attributes: i) the cumulative time spent by the passenger inside trains (T^{travel}), ii) the cumulative time spent by the passenger waiting between trains, in case of using multiple trains (T^{waiting}), iii) the number of times the passenger changes trains (m), iv) The difference between the desired departure time (t_p) and the actual one, noted by t , from the origin ($T^{\text{early}} = \max(0, t_p - t)$), if early and v) the time difference between the actual and the desired departure time from the origin, if late ($T^{\text{late}} = \max(0, t - t_p)$). The value of $U(j)$ is determined by Eq. (4).

$$U(j) = T^{\text{travel}} + \beta_1 T^{\text{waiting}} + m(\beta_2 + T^{\text{waiting}}) + \beta_3 T^{\text{early}} + \beta_4 T^{\text{late}} \quad (4)$$

We assume that passengers leave the system if they cannot reach their destination within a deadline. The deadline depends on the OD pair of passengers and is defined as a generalized travel time on the shortest path between o_p and d_p in the original timetable, plus a cutoff time κ . For each passenger, such an itinerary is created showing the opt-out option as the passenger can choose another transport mode. We assume that passengers are aware of the current timetable. Therefore, among all possible itineraries, the passenger chooses the one with the highest utility.

6.2. Passengers' priority lists

The passengers' priority list (\mathcal{L}) is an ordered set showing the passengers' assigning order (importance). The importance of the passenger $p \in P$ is shown by $\eta_p = v_p + \varepsilon_p$ in which v_p is the deterministic part of the importance function and ε_p is the stochastic part. The deterministic part imposes some rules on passengers' priority order (e.g., passengers with earlier desired departure time from the origin are prioritized, or passengers with longer travel time have priority to board the train). The stochastic part captures the elements that are not explicitly considered by the deterministic part. For example, the exact arrival time of passenger p or the platform's distance to the train door. For passenger p and passenger p' , if $\eta_p \geq \eta_{p'}$, then passenger p has boarding priority over passenger p' . There are various ways to define η_p , as our passengers are free to choose their itinerary, we assume that $v_p = 0$ and passengers are prioritized by the value of ε_p .

6.3. Passenger assignment graph

We present the passenger assignment by the time–space graph. The set of nodes is defined by $V' = S \times H \times K$ and the set of arcs is presented by $A = A_{\text{Dri}} \cup A_{\text{Wai}} \cup (\cup_{p \in P} A_{\text{Acc}}^p) \cup (\cup_{p \in P} A_{\text{Egr}}^p) \cup A_{\text{Tra}} \cup A_{\text{Opt}}$. The detailed description of arc sets and their associated costs and capacities are presented in Table 3. In this table, o_p and d_p are the nodes corresponding to the origin and destination stations of passenger p . Other nodes in time–space graph are noted by a tuple (s, t, k) in which s, t and k are indices for the station, time and train. The parameters related to the cost are described in Section 8.1 and q_k is the capacity of train k .

Finally, at each iteration, the passenger graph is adapted to the changes made to the timetable. The itinerary choice and the associated satisfaction can be easily determined by finding the shortest path on each passenger's graph. We use the Dijkstra algorithm, Dijkstra (1959), to find the shortest path.

Table 3
Arc weights and capacities.

Name	Set	Start node	End node	Cost	Capacity
Driving	A_{Dri}	(s, t, k)	(s', t', k)	$t' - t$	q_k
Waiting	A_{Wai}	(s, t, k)	(s, t', k)	$\beta_1 \cdot (t' - t)$	q_k
Access	A_{Acc}^p	o_p	(s, t, k)	$\beta_3 \cdot \max(0, (t_p - t)) + \beta_4 \cdot \max(0, (t - t_p))$	1
Egress	A_{Egr}^p	(s, t, k)	d_p	0	1
Transfer	A_{Tra}	(s, t, k)	(s, t', k')	$\beta_2 + (t' - t)$	∞
Opt-out	A_{Opt}	o_p	d_p	$t(o_p, d_p) + \kappa$	∞

6.4. Passenger assignment model

The passenger assignment model introduced by Binder et al. (2017a) relies on simulation to handle the stochasticity involved in defining the exogenous passengers' order. Their extensive numerical tests reveal that the variance across realizations of aggregate passenger satisfaction (e.g., total generalized travel time) is very low. Therefore, we assume that one iteration of the passengers' assignment model provides sufficient approximation for passenger flow and convenience for computational efficiency. Moreover, for consistency, the priority list is constructed in advance and is the same for all timetables. When all disposition timetables are created, one can use several iterations of the passengers' assignment model, as described in Binder et al. (2017a), to have a more precise outlook on passengers' satisfaction. Algorithm 3 presents the steps of the stochastic-based passenger assignment model. The algorithm performs several iterations till the set \mathcal{L} (i.e., priority list) is empty. At every iteration, the passenger with the highest priority is selected and removed from the list. Then the shortest path is calculated for the chosen passenger (noted by i_p). Based on the calculated path, the aggregate flow on each arc, indicated by $f(a)$, is updated. If the flow on arc a surpasses the train capacity, the arc will be removed from the graph in the next iteration. In the end, all passengers' itineraries, \mathcal{I} , returned.

Algorithm 3. Passengers' flow assignment

```

Input : Timetable ( $G$ )
          Passengers' list ( $P$ )
          Passengers' priority list  $\mathcal{L}$ 
Output : Passenger flows  $\mathcal{I}$ 
1 while  $O \neq \phi$  do
2    $p \leftarrow$  Select the passenger with the highest priority and remove it
   from  $\mathcal{L}$ .
3    $i_p \leftarrow$  Calculate the passengers' shortest path
4    $\mathcal{I} \leftarrow i_p$ 
5   foreach  $a \in i_p$  do
6      $f(a) = f(a) + 1$ 
7     if  $a \in A_{Dri} \cup A_{Wai}$  then
8       if  $f(a) \geq q_k$  then
9          $A \leftarrow A \setminus \{a\}, \forall p \in O$ 
10 return  $\mathcal{I}$ 

```

7. Objectives evaluations and archiving

By having a new disposition timetable G and passengers' itinerary \mathcal{I} , the algorithm then determines the objectives' value, archives the solution, and updates the operators' selection probability. These steps are summarized in Algorithm 4 and are described in the following sub-sections.

7.1. Objective value

We consider three objectives to evaluate the timetable's performance (i.e., operating cost, passengers convenience, and deviation from the original timetable). For solution, $X' = (G, \mathcal{I})$, the details of our calculation are presented as follows:

Operating cost. For the itinerary of train k , shown by r_k , the operating cost c_k is expressed as kilometer cost of running the train. The total operating cost is then $z_o(X') = \sum_{k \in K} c_k r_k$.

Passengers' convenience. For passenger p , the utility of the selected itinerary is noted by $U(i_p)$. The total passengers' satisfaction is then calculated by $z_{\mathcal{P}}(X') = \sum_{p \in P} U(i_p)$.

Deviation cost. We present the train itinerary (r_k) as an ordered-set of visiting nodes. That is $r_k = \{s^b, \dots, s', \dots, s^e\}$ in which s^b and s^e are the first and the last visited stations, respectively. For station s' , functions $t_k(s')$ and $t_k^*(s')$ return the departure time of train k in the disposition and original timetable. The duration of itinerary r_k is then calculated as $t_k(s^e) - t_k(s^b)$. To compute the deviation cost from the undisrupted timetable, four weighting factors are used: penalties for canceling (δ_c), delaying (δ_d) and rerouting trains (δ_r), and the cost of operating emergency trains (δ_e). The values of these penalties are presented in Section 8.1. We use the following steps to calculate the deviation cost:

- Train k has operated in the original timetable and it is canceled in the disposition timetable. The deviation cost (Δ_c^k) is proportional to the train duration in the original timetable. In other words, $\Delta_c^k = \delta_c \cdot (t_k^*(s^e) - t_k^*(s^b))$.

- Train k is scheduled as an emergency train. In this case, the deviation cost is calculated as a penalty cost of operating emergency train (δ_e) and the duration of train journey, that is $\Delta_e^k = \delta_e + t_k(s^e) - t_k(s^b)$.
- Train k has operated in the original timetable but its schedule and/or visiting stations are different. In this case, we first identify the last station until which the train follows its original planned journey and name it s' . Up to s' , the deviation cost is computed as delay cost. In other words, the delay cost is $\Delta_d^k = \delta_d \cdot \sum_{i \in \{s^b, \dots, s'\}} (t_k(i) - t_k^*(i))$. After station s' , there are three possibilities. i) If the train is partially canceled after s' , the deviation cost is $\Delta_c^k = \delta_c \cdot (t_k^*(s^e) - t_k(s'))$. ii) If the train is rerouted after s' , the deviation cost is $\Delta_r^k = \delta_r \cdot (t_k(s^e) - t_k(s'))$. iii) if the train is turned around after s' , we consider that an emergency train needs to be scheduled to compute the cost: $\Delta_e^k = \delta_e + (t_k(s^e) - t_k(s'))$.

Given the definitions above, the overall deviation cost is the sum over all trains and all types of deviations: That is $z_{\mathcal{D}}(X') = \sum_{k \in K} \Delta_c^k + \Delta_d^k + \Delta_r^k + \Delta_e^k$.

7.2. Archiving and acceptance criteria

After finding a new solution, we follow the strategy proposed by Suppapitnarm et al. (2000) for archiving. Let X and X' be two solutions. We say that X dominates X' if,

$$z_i(X) \leq z_i(X'), \forall i \in \{\mathcal{P}, \mathcal{O}, \mathcal{D}\} \quad \text{and} \quad \exists i \in \{\mathcal{P}, \mathcal{O}, \mathcal{D}\} : z_i(X) < z_i(X'),$$

where $z_{\mathcal{P}}, z_{\mathcal{O}}, z_{\mathcal{D}}$ are defined in Section 7.1. We denote this property by $X \prec X'$. A solution $X^* \in \mathcal{A}$ is said to be Pareto optimal if it is not dominated by any other solution: $\nexists X \in \mathcal{A} : X \prec X^*$. The set of all Pareto optimal solution is $\{X^* \in \mathcal{A} \mid \nexists X \in \mathcal{A} : X \prec X^*\}$. At every iteration, the current solution X' is a candidate for archiving. The following cases need to be considered:

- X' is dominated by at least one solution in \mathcal{A} (i.e., $\exists X \in \mathcal{A} : X \prec X'$). X' is not added to the archive and \mathcal{A} is not modified.
- X' is not dominated by any solution in \mathcal{A} (i.e., $\nexists X \in \mathcal{A} : X \prec X'$). Here, X' is added to \mathcal{A} , and all solutions in \mathcal{A} that are dominated by X' are removed from the archive.

Given this archiving procedure, we define the following acceptance criterion for the current solution X' . If X' is archived, it is automatically accepted. If not, it is accepted with a probability defined as follows:

$$\prod_{i \in \{\mathcal{P}, \mathcal{O}, \mathcal{D}\}} \min \left\{ \exp \left(- \frac{z_i(X') - z_i(X)}{T_i} \right), 1 \right\}. \tag{5}$$

This acceptance probability is the product of individual acceptance probabilities for each objective, with its associated temperature T_i . Note that the individual acceptance probability, $\exp(- (z_i(X') - z_i(X)) / T_i)$, may be greater than unity (if the difference is negative). We therefore take the minimum between the calculated value and one.

Algorithm 4. Archiving

```

Input : Solution  $X'$ 
        Archive  $\mathcal{A}$ 
Output : Update  $\mathcal{A}$ 
1 Evaluate  $z_{\mathcal{P}}(X'), z_{\mathcal{O}}(X'), z_{\mathcal{D}}(X')$  (Section 7.1)
2 if  $X'$  can be archived (Section 7.2) then
3   | Add  $X'$  to  $\mathcal{A}$ ;  $X \leftarrow X'$ 
4   | Update the score of  $\pi$  by  $\rho_1$  (Section 7.4)
5 else
6   | if  $X'$  is accepted by the SA criterion (Eq (5)) then
7     | |  $X \rightarrow X'$ 
8     | | Update the score of  $\pi$  by  $\rho_2$  (Section 7.4)
9   | else
10  | | Update the score of  $\pi$  by  $\rho_3$  (Section 7.4)

```

7.3. Temperature update

In this problem, a separate temperature is considered for every objective and note it by $T_i, i \in \{\mathcal{P}, \mathcal{O}, \mathcal{D}\}$. The temperatures, $T_i^{start} > 0$, are initially set to a sufficiently high value, in order to allow for a broad exploration of the search space, and to escape local

minima. The temperature is then gradually decreased as the search intensifies around a minimum. The cooling of the temperatures needs to be related to the objective functions' typical values to obtain meaningful acceptance probabilities. To that end, Suppapitnarm et al. (2000) suggest to compute after a predetermined number of iterations N_1 , the standard deviation of the respective objective functions, $\sigma(z_i)$, over all previously accepted solutions. The authors suggest $N_1 = 1000$ iterations, but this value can be reduced if the standard deviations stabilize earlier. After this warm-up phase, the temperatures are lowered M times, according to the following schedule:

$$T_i(m) = -\frac{\sigma(z_i)}{\ln(p_0 + \frac{p_f - p_0}{M}m)}, \quad m = 0, \dots, M. \quad (6)$$

This cooling strategy has the advantage that the acceptance probability decreases linearly, with respect to m , between the initial (p_0) and the final (p_f) acceptance probability. The algorithm is run \mathcal{R} times at this temperature, before the next cooling step.

7.4. Operators' selection and weight update

In Section 5.2 we have introduced 35 operators to modify the timetable. These operators are selected probabilistically. The operator selection probability depends on the weight assigned to each of them. In Algorithm 1 we define a segment as a fixed number of iterations during which the weights are updated before they are reset at the beginning of the next segment. With this approach, we can present the search process with a set of segments $j \in J$. For operator π in segment $j \in J$, we define three variables to adjust its selection probability dynamically. i) w_π^j shows the weight of the operator π at segment j , ii) q_π shows the score of the operator in each segment, iii) φ_π shows the number of times that operator π is used during a segment. In the following part, we first describe how the weights for general and infeasibility operators are updated. Then we describe how these operators are selected during the search process.

Weight update. For the general operators, we initialize the value of q_π and φ_π to zero at the beginning of each segment. Then, at every iteration, the score of the selected operator is updated. We distinguish three cases: i) if the last iteration results in a new solution is added to the archive (Algorithm 4, line 4), then the operator is rewarded with a score $\rho_1 > 0$, ii) if the last iteration results in a solution that was not added to the archive but was accepted by the simulated annealing criterion, the operator score is increased by $\rho_2 < \rho_1$, iii) if the last iteration results in a solution that was neither added to the archive nor accepted by the simulated annealing criterion, the score of the selected operator stays unchanged ($\rho_3 = 0$).

At the beginning of the algorithm, all weights are initialized at one ($w_\pi^0 = 1 \quad \forall \pi \in \Pi$). At the end of every segment (line 8, Algorithm 1), the weights are updated using the recorded scores in the following manner. After segment j , the weight of operator π in segment $j + 1$, w_π^{j+1} , is given by $w_\pi^{j+1} = (1 - \zeta)w_\pi^j + \zeta(q_\pi/\varphi_\pi)$. The reaction factor, ζ is a parameter that is defined between 0 and 1. This parameter controls the weight adjustment of the operators to keep them effective. If $\varphi_\pi = 0$, we assume that the weight of operator π remains unchanged in segment $j + 1$. A lower bound w_{\min} is enforced on weights. Its purpose is to avoid the probability of choosing an operator to become too small. Therefore, its value needs to be adjusted a posteriori once all operators' maximal weights are known.

For the infeasibility operator, the score is increased by ρ_1 if it produces a feasible solution. When the solution is not feasible but the operator manages to reduce the total number of infeasibilities, its score is increased by ρ_2 (line 8, Algorithm 2). Here we have to note that we do not use segments to update the weights of infeasibility operator. Therefore, their weights are updated once all infeasibilities of a solution are resolved (line 10, Algorithm 2), and the scores are set to zero.

Operators' selection. For general operators $\pi \in \Pi$, a two-level roulette wheel mechanism governs their selection process. In the first level, a set of operators is selected from $\Pi = \{\Pi_c, \Pi_D, \Pi_A, \Pi_I, \Pi_B\}$. The selection probability is the same for all sets in Π , thus ensuring fair use of cancellation, insertion, delay, advance, and bus insertion operators. In the second level, given the selected set (noted by Π'), operator $\pi' \in \Pi'$ is chosen with probability $w_{\pi'}^j / \sum_{\pi \in \Pi'} w_\pi^j$.

A single-level roulette-wheel mechanism is implemented for infeasibility operators ($\pi \in \Pi_{Inf}$). In this case, the selection only depends on the operators' weight. In the next subsection, we explain our strategy to intensify the search process.

7.5. Return-to-archive strategy

To explore the search space close to the Pareto frontier more intensively, we implement an occasional selection of a solution in the archive. The rate at which to come back to a solution from the archive is decided in the following way. In the warm-up phase, before the iteration count hits N_1 , we do not perform any return-to-archive moves, as the goal is to explore the search space as freely as possible at that time. After that, the rate of return to solutions of the archive is increased in the following manner: $N_j^B = r_B \cdot N_{j-1}^B$ $j = 2, 3, 4, \dots$, where N_j^B is the number of iterations to be executed prior to the j th return-to-archive and $0 < r_B < 1$ is a parameter controlling the rate of return. Suitable values of the parameters found by Suppapitnarm et al. (2000) are $r_B = 0.9$ and $N_1^B = 2K$. The selection of the solution from archive is performed in a random manner.

8. Computational experiments

The algorithm is implemented in Java, and all numerical experiments have been carried out on a 3.33 GHz Intel Xeon X5680 server running 64-bit Ubuntu 16.04.2. Computational results are presented in the following manner. In Section 8.1, we briefly introduce the case studies on which the algorithm is applied. The contribution of different operators are discussed in Section 8.2, and the

computational results are presented in Section 8.3. Finally, the practical applicability of the framework is discussed in Section 8.4.

8.1. Description of case studies

We evaluate our proposed resolution algorithm on the Dutch and the Swiss rail networks. The Dutch case is used to calibrate the algorithm, and the Swiss case is used to evaluate our proposed approach. For the detailed description of these case studies, we refer the reader to Binder et al. (2017b) and Binder et al. (2017a). We create a large set of “small” instances from the Dutch case network to calibrate our algorithm’s parameters. The Dutch network comprises 18 double tracks between stations. For each track, the following disruptions are considered:

- complete blockade for the entire time horizon (0–120 min);
- complete blockade for the first half of the time horizon (0–60 min);
- partial blockade for the entire time horizon (0–120 min).

A total of 54 test instances are thus created. For each of instance, the algorithm is run, and results are reported in Section 8.2. For the Swiss case study, the network of S-trains in canton Vaud Switzerland is considered. Fig. 2 shows the network of S-train. For this case, we consider the recovery period between 5:00 am and 9:00 am. For this network, two disruption scenarios are considered. The first scenario is a full blockade of the tracks between stations Montreux (MON) and Vevey (VEV), between 05:30 am, and 06:30 am. The main purpose is to analyze the disposition timetables constructed by our framework and to validate the latter by comparing the algorithm’s output with the recovery procedures used in practice. The second scenario illustrates a more severe case, with the two following disruptions happening simultaneously:

- complete blockade of the tracks between Lausanne (LAU) and Vevey (VEV), from 5:00am to 8:00am;
- complete blockade of the tracks between Lausanne (LAU) and Pully (PUI), from 5:00am to 7:00am.

For the two scenarios, the algorithm is run ten times, and the archives of non-dominated solutions of every run are merged together, according to the dominance rules described in Section 7.2. Tables 4 and 5 summarize the values we use for the parameters of the algorithm. These values are either borrowed from the literature or calibrated based on preliminary trial-and-error combinations. The 54 instances of the Dutch case study are extensively used to calibrate the parameters.

8.2. Algorithm performance

In this section, we aim at showing our algorithm’s performance by discussing the functionality of operators. We report the results on 54 instances from the Dutch railway. First, the variation in the operators’ selection probability is examined in Fig. 3, both during and at the end of the algorithm. We discuss the operators’ contribution to the search for non-dominated solutions in Table 6. Then, the performance of the “infeasibility operator” is presented in Table 7. Finally, computational times associated with the operators are

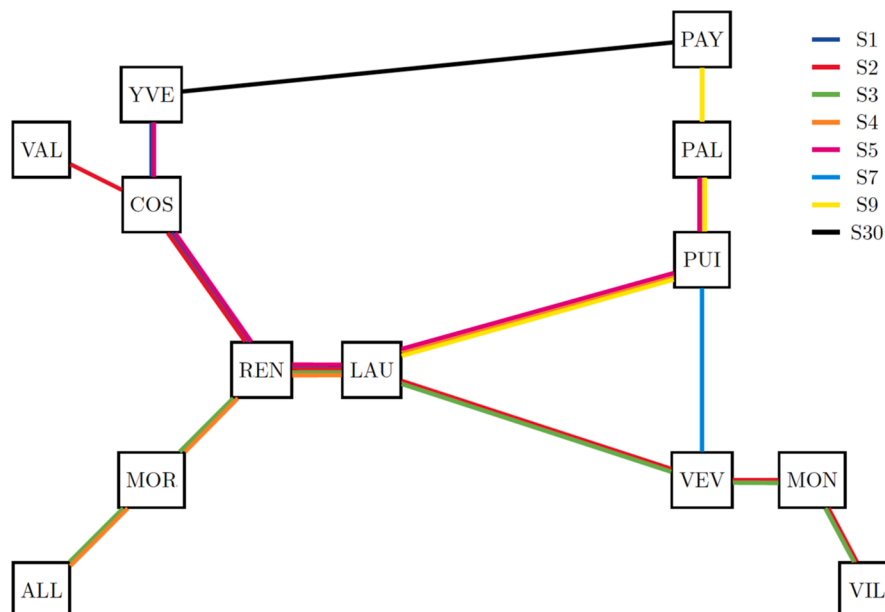


Fig. 2. Network of S-trains in canton Vaud, Binder et al. (2017a).

Table 4
Parameters' value.

Symbol	Description	Value	Source
T_1^{start}	Initial temperature	10^8	Trial-and-error
p_0	Initial acceptance probability	0.999	Bierlaire (2015)
p_f	Final acceptance probability	0.001	Bierlaire (2015)
M	Number of temperature changes	100	Bierlaire (2015)
\mathcal{N}	Iterations per temperature level	50	Bierlaire (2015)
N_1	Warm-up iterations	300	Trial-and-error
N_1^B	Iterations before first return-to-archive	2K	Suppaitnarm et al. (2000)
r_B	Decrease parameter	0.99	Suppaitnarm et al. (2000)
J	Weight segment length	2K	Trial-and-error
ζ	Reaction factor	0.5	Ropke and Pisinger (2006)
ρ_1	Reward	10	Trial-and-error
ρ_2	Reward	2	Trial-and-error
ρ_3	Reward	0	Trial-and-error
w_{min}	Minimal operator weight	0.3	Trial-and-error
ν_{max}	Maximal number of evaluated solutions	20	Trial-and-error
δ_d	Deviation cost for delayed train (per minute)	1	Veelenturf et al. (2016b)
δ_c	Deviation cost for cancelled train	50	Veelenturf et al. (2016b)
δ_r	Deviation cost for re-routing train	10	Veelenturf et al. (2016b)
δ_e	Penalty to lunch a new trains	$1000 + OC^1$	Veelenturf et al. (2016b)

1) OC: operating cost of the new train

Table 5
Values of weighting factors in the passengers' generalized travel time.

Parameter	Value	Unit	Reference
β_1	2.5	[min/min]	Wardman (2004)
β_2	10	[min/transfer]	de Keizer et al. (2012)
β_3	0.5	[min/min]	Small (1982)
β_4	1	[min/min]	Small (1982)

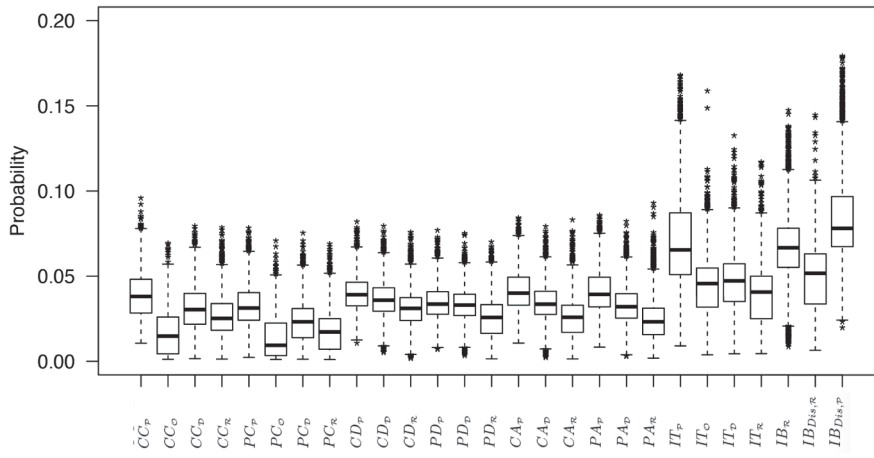
reported in Fig. 4.

Recall that during the search, the operators' weight is dynamically updated based on their past performance. This influences the selection probability of the operators. Moreover, after a fixed number of iterations (i.e., segment), we reset the weight associated with each operator. Here, we have to note that at the beginning of each segment, operators' selection probability is updated by the procedure described in Section 7.4. Fig. 3 reports the distribution of probabilities for general operators. Fig. 3a shows the distribution of selection probabilities (see Section 7.4) for all segments over 54 instances. In Fig. 3b we only report operators selection probability for the last segment of the search procedure.

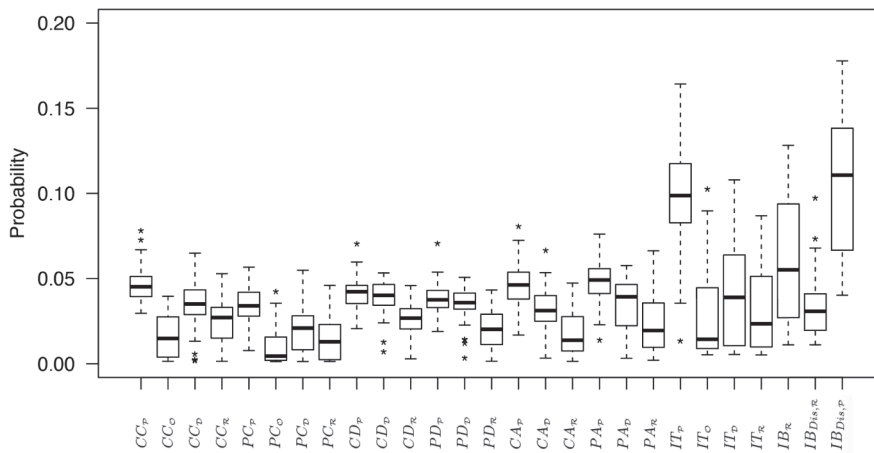
The probabilities reported in Fig. 3a indicate that each operator is useful at some point when considering results across all instances. Indeed, the median probability of selecting an operator is different from zero across all segments, and every operator exhibits high values of its maximal probability of being chosen. At the same time, the high variability of the probabilities to select an operator demonstrates the usefulness of the weight adaptation scheme. In other words, an increment in selection probability means that the operator weight is increased, which shows that the operator succeeds in improving the timetable. Even for operators such as CC_O , PC_O , PC_R , which exhibit very low final probabilities, there are instances where the probability is higher. An operator can therefore be very efficient in the case of one instance, but not contribute significantly to another, thus suggesting that the weight updating scheme is capable of adapting to the specificities of each instance.

We note that final probabilities (Fig. 3b) accentuate the overall trend observed during the algorithm (Fig. 3a). It can thus be observed which operators become essential and which are not that useful any more. Operators minimizing passenger inconvenience appear to have consistently higher probabilities of being selected for the groups of cancel, advance, train, and bus insertion operators ($CC_{\mathcal{P}}$, $PC_{\mathcal{P}}$, $CA_{\mathcal{P}}$, $PA_{\mathcal{P}}$, $IT_{\mathcal{P}}$ and $IB_{Dis,\mathcal{P}}$). Regarding the delay operators, there appears to be no clear preference between minimizing passenger inconvenience or the deviation from the undisturbed timetable (see $CD_{\mathcal{P}}$, $CD_{\mathcal{S}}$ and $PD_{\mathcal{P}}$, $PD_{\mathcal{S}}$).

Table 6 reports over the 54 test instances for each general operator: the number of times it found a non-dominated solution; the number of times it was used; and its success rate defined as the ratio between the two previous values. Note that the number of non-dominated solutions is an overestimation of the total number of solutions in the final archive, as a non-dominated solution in an early iteration might become dominated by a later solution. Each operator can find a significant amount of non-dominated solutions that corroborate the different operators' usefulness described above. Further, it can be observed that operators with low success rates (such



(a) Distribution of operators' selection probabilities at the end of all segments.



(b) Distribution of operators' selection probabilities at the end of the final segment.

Fig. 3. Distribution of selecting probabilities of each operator over the 54 test instances.

as IT_O, IT_D, IT_R, IB_R and $IB_{Dis,R}$) are not necessarily the ones with the lowest number of usages, indicating that those should not be discarded as useless, but rather considered as diversification opportunities that pave the path for a later significant improvement leading to a new non-dominated solution. Likewise, operators with a very high number of usages are not the ones with the best success rate.

Table 7 reports the usage statistics of infeasibility operators, as well as their capacity to restore feasibility. The number of resolved infeasibilities indicates how many times an operator could generate a feasible neighbor timetable. In contrast, the number of reduced infeasibilities reports the number of times the operator generated a neighbor timetable with a lower number of infeasibilities than the incumbent solution. One can observe that the complete cancelation operator is by far the most used, which is explained by the fact that resolving an infeasibility is highly rewarded in the algorithm's weighting scheme. The other infeasibility operators are nevertheless used frequently, except for the advance operators (CA_{Inf} and PA_{Inf}). The latter results from the nature of the considered disruptions: all disruptions start at the beginning of the time horizon, and it is, therefore, impossible to schedule a train to pass before the disruption happens. This demonstrates once more the usefulness of the weight adaptation scheme.

Finally, Fig. 4 reports the distribution of average computational times during a segment per operator and over 54 instances. Local search operators exhibit longer computational times than the operators with a different search strategy. Overall, computational times are very low, at less than one-tenth of a second per operator usage.

Table 6
Usage statistics and success rates of general operators.

General operator	# of non-dominated solutions found	# of usages	Success rate
$CC_{\mathcal{P}}$	4572	10,672	0.43
$CC_{\mathcal{E}}$	1229	4863	0.25
$CC_{\mathcal{D}}$	3119	8581	0.36
$CC_{\mathcal{R}}$	2113	7373	0.29
$PC_{\mathcal{P}}$	3016	9105	0.33
$PC_{\mathcal{E}}$	718	3941	0.18
$PC_{\mathcal{D}}$	1448	6357	0.23
$PC_{\mathcal{R}}$	923	5116	0.18
$CD_{\mathcal{P}}$	4077	11,108	0.37
$CD_{\mathcal{D}}$	3411	10,393	0.33
$CD_{\mathcal{E}}$	2121	8750	0.24
$PD_{\mathcal{P}}$	2715	9517	0.29
$PD_{\mathcal{D}}$	2626	9325	0.28
$PD_{\mathcal{R}}$	1360	7006	0.19
$CA_{\mathcal{P}}$	3508	11,424	0.31
$CA_{\mathcal{D}}$	1944	9644	0.20
$CA_{\mathcal{E}}$	446	3382	0.13
$PA_{\mathcal{P}}$	2876	9548	0.30
$PA_{\mathcal{D}}$	1500	7743	0.19
$PA_{\mathcal{E}}$	210	1228	0.17
$IT_{\mathcal{P}}$	4299	19,716	0.22
$IT_{\mathcal{E}}$	931	12,542	0.07
$IT_{\mathcal{D}}$	1118	13,310	0.08
$IT_{\mathcal{R}}$	661	10,894	0.06
$IB_{\mathcal{R}}$	317	18,504	0.02
$IB_{Dis,\mathcal{R}}$	179	13,892	0.01
$IB_{Dis,\mathcal{P}}$	1746	23,476	0.07

Table 7
Usage statistics of infeasibility operators.

Infeasibility operator	# of resolved infeasibilities	# of reduced infeasibilities	# of usages
CC_{inf}	19,587	725	20,312
PC_{inf}	3435	1021	4456
CD_{inf}	2250	232	5223
PD_{inf}	660	333	1664
CA_{inf}	0	0	236
PA_{inf}	0	1	13
RR_{inf}	3821	149	4798
TA_{inf}	2597	830	3795

8.3. Effectiveness in handling disruption

In the following section, we show the performance of our algorithm to handle large disruptions. For this reason, we have applied our algorithm to the Swiss case study described in Section 8.1.

8.3.1. Disruption MON–VEV

As introduced in Section 8.1, this section focuses on the analysis of the disposition timetables constructed by our framework for a disruption scenario that is easy to solve due to the limited rescheduling possibilities. Indeed, a full blockade of the tracks between stations MON and VEV from 5:30 am to 6:30 am is considered. In the normal timetable, three trains are scheduled on the disrupted tracks:

- Train **S2 12211**, leaving VEV at 06:21 am and reaching MON at 06:30 am;

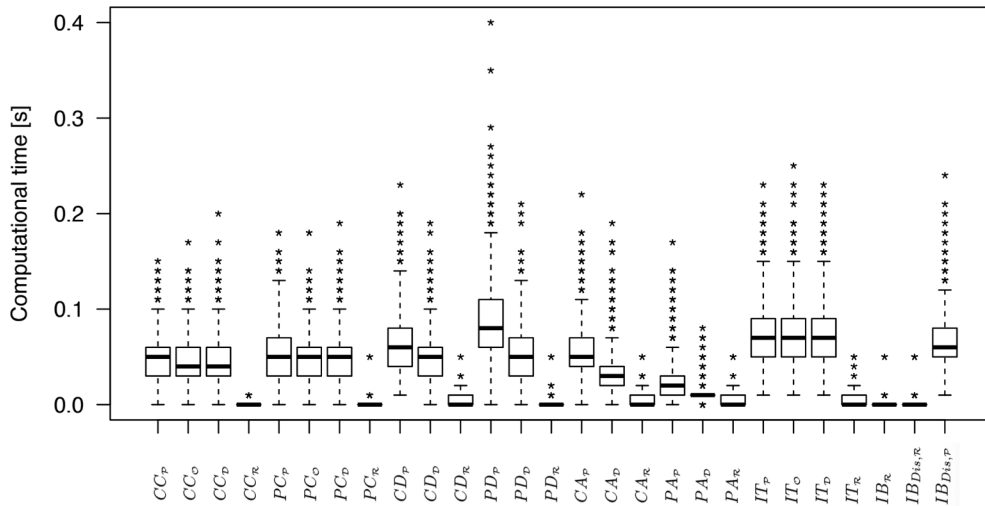


Fig. 4. Distribution of computational times of the operators over the 54 test instances.— Change notations.

- Train S2 12216, leaving MON at 06:29 am and reaching VEV at 06:37 am;
- Train S3 12314, leaving MON at 05:55 am and reaching VEV at 06:03 am.

For this disruption scenario, the merged archive of non-dominated solutions obtained by our framework contains 952 disposition timetables. Table 8 summarizes the recovery strategies applied to the three disrupted trains in each of the timetables: column CC indicates the number of timetables in which the train is completely canceled; column PC the number of timetables in which the train is partially canceled, and after which station; column CD the number of timetables in which the train is completely delayed; column PD the number of timetables in which the train is partially delayed, and after which station; and column TA the number of timetables in which the train is turned around, and after which station.

It can be observed that “common sense” recovery actions are properly reproduced by our algorithm. For instance, trains S1 12211 and S2 12216, which are scheduled in the normal timetable to cross the disrupted tracks only very shortly before the disruption ending time (9 and 1 min, respectively), are much more likely to be delayed (partially or completely) than train S3 12314. Indeed, by delaying these trains only by a few minutes, the timetable easily becomes feasible again. In particular, it is noteworthy that the algorithm proposes timetables in which these two trains are delayed so as to depart exactly at 06:30 am (the ending time of the disruption) from stations MON or VEV. Further, the most common station for partial cancelations and turn arounds is the station right before the disruption (VEV for the first train, MON for the two others), which makes sense from a practical point of view.

Conversely, the algorithm also produces less “intuitive” timetables, such as turning around train S2 12211 in station COS or delaying train S3 12314 by more than 35 min in order to pass the disrupted area. In addition to the recovery actions applied to the three trains, we examine the emergency buses and trains scheduled in the disposition timetables. On average, 5.8 buses are scheduled, and this number varies between 0 and 16, depending on the timetable. In 553 out of the 952 timetables, emergency trains are scheduled on the disrupted tracks after the disruption is resolved. Interestingly, no emergency train is scheduled before the beginning of the disruption at 05:30 am, thus showing the algorithm adapts to the low passenger demand at this time.

8.3.2. Disruption LAU-VEV-PUI

This section is dedicated to the discussion of the most severe disruption scenario introduced in Section 8.1. Results are presented at the aggregate level: timetables in the archive of non-dominated solutions are first compared using the Pareto frontier of the three objectives — passenger inconvenience (z_{ϕ}), operational cost (z_{ρ}), and deviation from the undisrupted timetable (z_{φ}). Then, the three

Table 8

Summary of recovery strategies applied to trains S2 12211, S2 12216 and S3 12314 in disruption MON-VEV.

Train	Recovery strategy usage				
	CC	PC	CD	PD	TA
S2 12211	440	14 (COS) 7 (LAU) 15 (MON) 5 (REN) 278 (VEV)	38	23 (LAU) 92 (VEV)	1 (COS) 39 (VEV)
S2 12216	337	363 (MON)	227	12 (MON)	13 (MON)
S3 12314	343	481 (MON)	1	11 (MON)	116 (MON)

timetables in the archive with the lowest value of $z_{\mathcal{D}}$, $z_{\mathcal{C}}$, and $z_{\mathcal{D}}$, respectively, are reported in Table 9 and compared in further detail. The complexity of the Pareto frontiers obtained by our framework requires a sound methodology for practitioners to choose which timetable to implement among the numerous non-dominated solutions. Two of such methodologies are proposed in Section 8.4. For this disruption scenario, the merged archive of non-dominated solutions contains 987 disposition timetables. Fig. 5 compares each of the timetables' performance in terms of the objective functions identified above. As the Pareto frontier is three-dimensional, we project it on the three planes ($z_{\mathcal{D}} = 0$, $z_{\mathcal{C}} = 0$, $z_{\mathcal{C}} = 0$) for an easier visual interpretation, and the third objective is represented in greyscale.

It can be observed in Fig. 5a that the starkest trade-off between objectives is achieved on the $z_{\mathcal{C}}-z_{\mathcal{D}}$ front: an increase in operational cost results (almost) automatically in a decrease in passenger inconvenience, regardless of the value of $z_{\mathcal{D}}$. Also, by increasing the deviation from the undisrupted timetable, the two other objectives can be improved simultaneously. Note that it is not possible to identify a clear inflection point of the frontier. This may be explained by the fact that the cost of the opt-out arc has been lowered, thus effectively limiting the sharp increase in passenger inconvenience when canceling trains (i.e., decreasing operational cost).

The trade-off in the two other frontiers is less clear. In the $z_{\mathcal{C}}-z_{\mathcal{D}}$ projection (Fig. 5b), the minimal value of deviation from the undisrupted timetable lies at about $z_{\mathcal{C}} = 70$ k CHF. From this point, $z_{\mathcal{D}}$ can be increased in two ways: either by decreasing operational cost (thus increasing passenger inconvenience), or vice versa. The former situation corresponds to timetables where many trains are canceled, resulting in poor passenger satisfaction, whereas in the latter additional trains and buses are scheduled, increasing passenger satisfaction but coming at a cost. Finally, in the $z_{\mathcal{D}}-z_{\mathcal{D}}$ projection (Fig. 5c), it can be observed that, for a given value of operational cost, an increase in $z_{\mathcal{D}}$ results in a decrease in passenger inconvenience. This indicates that the operators introduced in our framework are capable of scheduling trains that are adjusted to passenger demand.

Table 9 reports the values of the objective functions at the minimal extremes of the Pareto frontier, as well as the level of unsatisfied demand, defined as the number of passengers that need to take the opt-out path. For comparison, in the undisrupted timetable, $z_{\mathcal{D}} = 901$, 924.5 [min], $z_{\mathcal{C}} = 92$, 160.0 CHF and $z_{\mathcal{D}} = 0$. For the timetable with the lowest passenger inconvenience (first row of Table 9), $z_{\mathcal{D}}$ is only about 4% higher than in the undisrupted case. This level of service however, requires a high operational cost. Indeed, 18 buses and 19 emergency trains are scheduled in this timetable. Even in this case, 2155 out of 14,920 passengers (14%) take the opt-out path. The timetable with minimal operational cost (second row of Table 9) is associated with an extremely low level of service for passengers: on average, travel times increase by about 64%, and more than two-thirds of the passengers need to find an alternative solution to travel. This timetable is "cheap" for the railway operator because only 30 original and six emergency trains are operated. In addition, all emergency trains run on train lines with at most two stations. The deviation cost associated with this timetable is very high because many original trains are completely canceled. Finally, the timetable with the lowest deviation cost (third row of Table 9) represents an interesting compromise: $z_{\mathcal{D}}$ increases by 18% concerning the undisrupted case and the associated operational cost remains low. Only one bus is scheduled in this timetable, and not a single emergency train. Railway operators could therefore use this timetable as an initial feasible solution, and then include additional trains or buses to increase the level of service provided, especially to serve the unsatisfied demand.

8.4. Practical applicability

In conclusion, the results show that the proposed framework can construct high-quality feasible disposition timetables that make sense from a practical point of view. Also, as can be seen in Fig. 6, the computational time of the algorithm remains low, even for the real case study based on the Swiss S-train network. The highest computational times occur with the local search operators, mostly those minimizing passenger inconvenience, as the passenger assignment model needs to be rerun on each timetable in this particular case. On average, it takes 0.2 s for the passenger assignment model to determine the passengers' flow.

Therefore, the algorithm described in this paper can be seen as a tool for railway operators to compare disposition timetables at an aggregated level. Its output is a list of timetables that are non-dominated. To choose the "best" among them, train operators wishing to use the framework need to decide the relative importance of the objectives to their eyes or the maximal value of each objective that they are willing to accept. If one objective is more important than the others, the operator will choose a timetable that minimizes this objective while respecting the other objectives' bounds. Otherwise, if the objectives are equivalently important, the closest timetable (e.g., in terms of Euclidian distance in the three-dimensional Pareto frontier) to the undisrupted timetable is preferred.

We wish to stress here that each timetable in the archive of non-dominated solutions results from successive applications of recovery operators that are directly inspired by practice. Therefore, it is practical for railway operators to implement a timetable from the archive by retracing the elementary steps that led to it. Finally, note that operator-defined upper bounds on the objective functions can easily be included in the framework. This has the advantage of limiting the search for timetables to a solution space in which operators are guaranteed to obtain timetables in line with their requirements.

9. Conclusion

This paper introduces an algorithmic framework to design the disposition timetable, taking into account train's capacity. Our algorithm is capable of providing a user equilibrium solution for the disposition timetable. We consider operating cost, deviation from the original timetable, and passengers' convenience as the main three performance indicators. The algorithm constructs an approximation of the Pareto frontier of the problem by populating an archive of non-dominated solutions.

We test our algorithm on the case studies from Dutch and Swiss railway. Our analysis indicates that each of the operators contributes at some point to find new non-dominated solutions. Additionally, it is shown that the framework performs well concerning computation time and solution quality compared to practice in many cases. We have presented an approach on how to use our

Table 9

Values of the objective functions, and level of unsatisfied demand, for the timetables with lowest value of z_p, z_o and z_d in \mathcal{A} .

z_p [min]	z_o [CHF]	z_d [-]	Unsatisfied demand
934,290.0	104,970.0	67,883.0	2155
1,486,869.5	32,850.0	109,497.0	10,526
1,069,571.5	71,670.0	31,716.0	3524

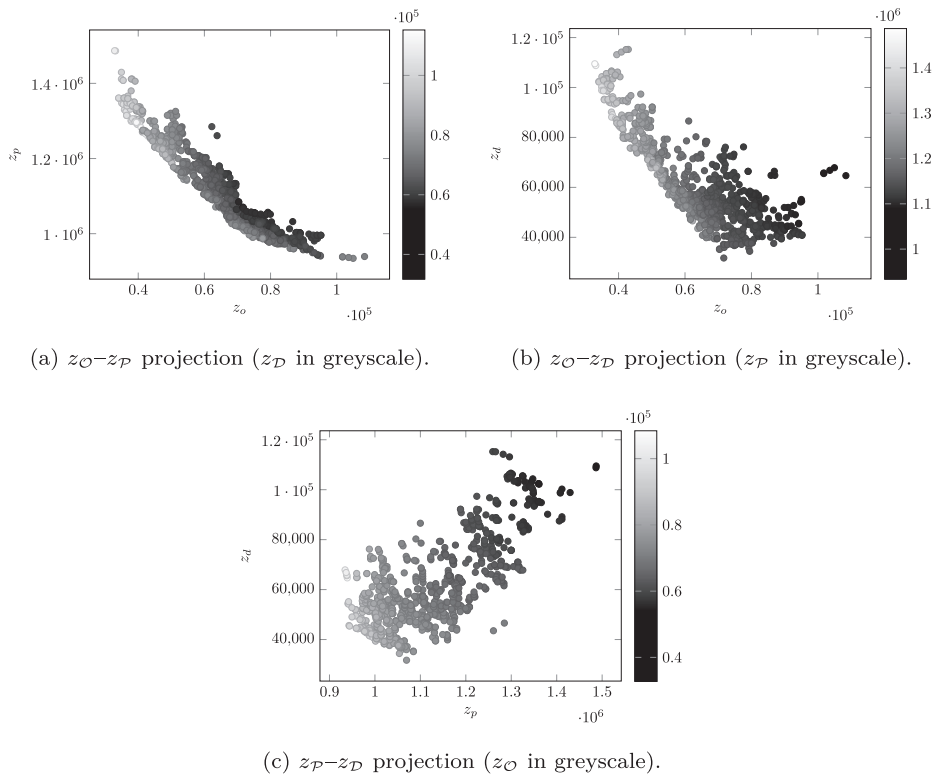


Fig. 5. Pareto frontier projections for disruption LAU-VEV-PUI.

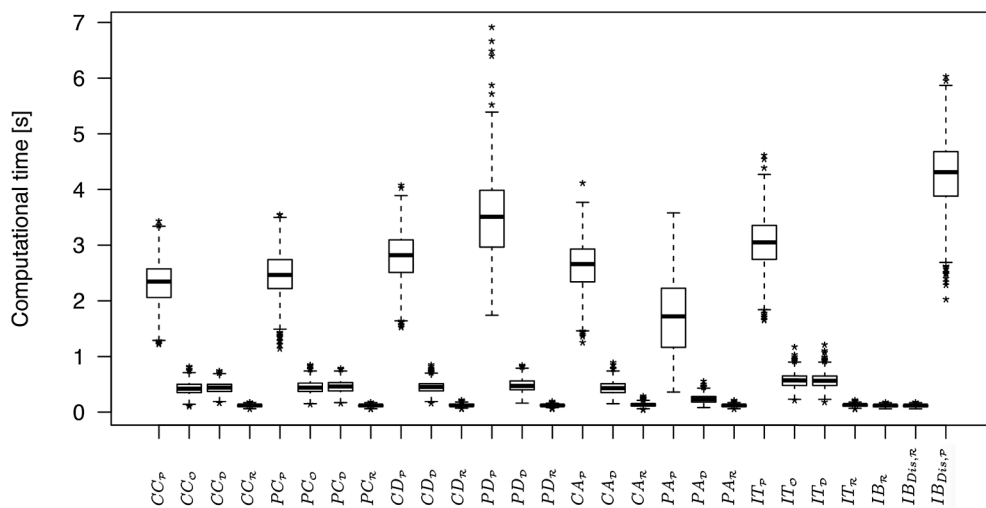


Fig. 6. Distribution of computational times for disruption LAU-VEV-PUI.

algorithms in practice. In this problem, we assume that passengers know the current disposition timetable and choose their routes accordingly. Their choice can be influenced by the route advice that they can receive from the train operating company. As future work, we are interested in investigating how to incorporate route advice on planning and designing the disposition timetable.

Declaration of Competing Interest

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.

References

- Albrecht, A., Pantou, D., Lee, D., 2013. Rescheduling rail networks with maintenance disruptions using Problem Space Search. *Computers & Operations Research* 40, 703–712.
- Almodóvar, M., García-Ródenas, R., 2013. On-line reschedule optimization for passenger railways in case of emergencies. *Computers & Operations Research* 40, 725–736.
- Batley, R., Dargay, J., Wardman, M., 2011. The impact of lateness and reliability on passenger rail demand. *Transportation Research Part E: Logistics and Transportation Review* 47, 61–72.
- Benezech, V., Coulombel, N., 2013. The value of service reliability. *Transportation Research Part B: Methodological* 58, 1–15.
- Bierlaire, M., 2015. *Optimization: Principles and Algorithms*. EPFL Press, Lausanne.
- Binder, S., Maknoon, Y., Bierlaire, M., 2017a. Exogenous priority rules for the capacitated passenger assignment problem. *Transportation Research Part B: Methodological* 105, 19–42.
- Binder, S., Maknoon, Y., Bierlaire, M., 2017b. The multi-objective railway timetable rescheduling problem. *Transportation Research Part C: Emerging Technologies* 78, 78–94.
- Brucker, P., Heitmann, S., Knust, S., 2002. Scheduling railway traffic at a construction site. *OR Spectrum* 24, 19–30.
- 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. *Transportation Research Part B: Methodological* 63, 15–37.
- Cadarso, L., Marin, A., Maroti, G., 2013. Recovery of disruptions in rapid transit networks. *Transportation Research Part E: Logistics and Transportation Review* 53, 15–33.
- Chen, Z., Li, X., Zhou, X., 2019. Operational design for shuttle systems with modular vehicles under oversaturated traffic: Discrete modeling method. *Transportation Research Part B: Methodological* 122, 1–19.
- Corman, F., 2020. Interactions and equilibrium between rescheduling train traffic and routing passengers in microscopic delay management: a game theoretical study. *Transportation Science* 54, 785–822.
- Corman, F., D'Ariano, A., Hansen, I.A., Pacciarelli, D., 2011. Optimal multi-class rescheduling of railway traffic. *Journal of Rail Transport Planning & Management* 1, 14–24.
- Corman, F., D'Ariano, A., Marra, A.D., Pacciarelli, D., Samà, M., 2016. Integrating train scheduling and delay management in real-time railway traffic control. *Transportation Research Part E: Logistics and Transportation Review*.
- Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M., 2012. Bi-objective conflict detection and resolution in railway traffic management. *Transportation Research Part C: Emerging Technologies* 20, 79–94.
- Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M., 2014. Dispatching and coordination in multi-area railway traffic management. *Computers & Operations Research* 44, 146–160.
- D'Ariano, A., Pacciarelli, D., Pranzo, M., 2007. A branch and bound algorithm for scheduling trains in a railway network. *European journal of operational research* 183, 643–657.
- Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. *Numer. Math.* 1, 269–271.
- Dollevoet, T., Huisman, D., Schmidt, M., Schöbel, A., 2012. Delay Management with Rerouting of Passengers. *Transportation Science* 46, 74–89.
- Espinosa-Aranda, J.L., García-Ródenas, R., 2013. A demand-based weighted train delay approach for rescheduling railway networks in real time. *Journal of Rail Transport Planning & Management* 3, 1–13.
- Ghaemi, N., Cats, O., Goverde, R.M., 2018. Macroscopic multiple-station short-turning model in case of complete railway blockages. *Transportation Research Part C: Emerging Technologies* 89, 113–132.
- van der Hurk, E., Kroon, L., Maroti, G., 2018. Passenger advice and rolling stock rescheduling under uncertainty for disruption management. *Transportation Science* 52, 1391–1411.
- Kanai, S., Shiina, K., Harada, S., Tomii, N., 2011. An optimal delay management algorithm from passengers' viewpoints considering the whole railway network. *Journal of Rail Transport Planning & Management* 1, 25–37.
- de Keizer, B., Geurs, K., Haarsman, G., 2012. Interchanges in timetable design of railways: A closer look at customer resistance to interchange between trains, in: 40th European Transport Conference, Glasgow.
- Kroon, L., Maroti, G., Nielsen, L., 2015. Rescheduling of railway rolling stock with dynamic passenger flows. *Transportation Science* 49, 165–184.
- Louwerse, I., Huisman, D., 2014. Adjusting a railway timetable in case of partial or complete blockades. *Eur. J. Oper. Res.* 235, 583–593.
- Meng, L., Zhou, X., 2014. Simultaneous train rerouting and rescheduling on an n-track network: A model reformulation with network-based cumulative flow variables. *Transportation Research Part B: Methodological* 67, 208–234.
- Narayanaswami, S., Rangaraj, N., 2013. Modelling disruptions and resolving conflicts optimally in a railway schedule. *Computers & Industrial Engineering* 64, 469–481.
- Niu, H., Zhou, X., 2013. Optimizing urban rail timetable under time-dependent demand and oversaturated conditions. *Transportation Research Part C: Emerging Technologies* 36, 212–230.
- Parbo, J., Nielsen, O.A., Prato, C.G., 2016. Passenger perspectives in railway timetabling: a literature review. *Transport Reviews* 36, 500–526.
- Pellegrini, P., Marlière, G., Rodriguez, J., 2014. Optimal train routing and scheduling for managing traffic perturbations in complex junctions. *Transportation Research Part B: Methodological* 59, 58–80.
- Ropke, S., Pisinger, D., 2006. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science* 40, 455–472.
- Sato, K., Tamura, K., Tomii, N., 2013. A mip-based timetable rescheduling formulation and algorithm minimizing further inconvenience to passengers. *Journal of Rail Transport Planning & Management* 3, 38–53.
- Schachtebeck, M., Schöbel, A., 2010. To wait or not to wait - and who goes first? delay management with priority decisions. *Transportation science* 44, 307–321.
- Schöbel, A., 2001. A Model for the Delay Management Problem based on Mixed-Integer-Programming. *Electronic Notes in Theoretical Computer Science* 50, 1–10.
- Schöbel, A., 2009. Capacity constraints in delay management. *Public Transport* 1, 135–154.
- Small, K.A., 1982. The scheduling of consumer activities: Work trips. *Am. Econ. Rev.* 72, 467–479.
- Suppattitnarm, A., Seffen, K.A., Parks, G.T., Clarkson, P.J., 2000. A simulated annealing algorithm for multi-objective optimization. *Engineering Optimization* 33, 59–85.

- Tian, X., Niu, H., 2020. Optimization of demand-oriented train timetables under overtaking operations: A surrogate-dual-variable column generation for eliminating indivisibility. *Transportation Research Part B: Methodological* 142, 143–173.
- Törnquist, J., Persson, J.A., 2007. N-tracked railway traffic re-scheduling during disturbances. *Transportation Research Part B: Methodological* 41, 342–362.
- Riden de Treinen, R.d.T., 2021. Disruption statistics. URL: <https://www.rijdendetreinen.nl/en/statistics>.
- Veelenturf, L.P., 2014. Disruption management in passenger railways: models for timetable, rolling stock and crew rescheduling. Ph.D. thesis. Erasmus Universiteit Rotterdam. Rotterdam.
- Veelenturf, L.P., Kidd, M.P., Cacchiani, V., Kroon, L., Toth, P., 2016a. A railway timetable rescheduling approach for handling large-scale disruptions. *Transportation Science* 50, 841–862.
- Veelenturf, L.P., Kidd, M.P., Cacchiani, V., Kroon, L.G., Toth, P., 2016b. A Railway Timetable Rescheduling Approach for Handling Large-Scale Disruptions. *Transportation Science* 50, 841–862.
- Wardman, M., 2004. Public transport values of time. *Transp. Policy* 11, 363–377.
- Zhan, S., Kroon, L., Veelenturf, L.P., Wagenaar, J.C., 2015. Real-time high-speed train rescheduling in case of a complete blockage. *Transportation Research Part B: Methodological* 78, 182–201.
- Zhan, S., Kroon, L.G., Zhao, J., Peng, Q., 2016. A rolling horizon approach to the high speed train rescheduling problem in case of a partial segment blockage. *Transportation Research Part E: Logistics and Transportation Review* 95, 32–61.
- Zhan, S., Wong, S., Shang, P., Peng, Q., Xie, J., Lo, S., 2021. Integrated railway timetable rescheduling and dynamic passenger routing during a complete blockage. *Transportation Research Part B: Methodological* 143, 86–123.
- Zhu, Y., Goverde, R.M., 2019. Railway timetable rescheduling with flexible stopping and flexible short-turning during disruptions. *Transportation Research Part B: Methodological* 123, 149–181.
- Zhu, Y., Goverde, R.M., 2020a. Dynamic and robust timetable rescheduling for uncertain railway disruptions. *Journal of Rail Transport Planning & Management* 100–196.
- Zhu, Y., Goverde, R.M., 2020b. Integrated timetable rescheduling and passenger reassignment during railway disruptions. *Transportation Research Part B: Methodological* 140, 282–314.
- Zhu, Y., Goverde, R.M., 2021. Dynamic railway timetable rescheduling for multiple connected disruptions. *Transportation Research Part C: Emerging Technologies* 125, 103080.