Integrated timetable rescheduling and passenger reassignment during railway disruptions

Journal Article

Author(s):
Zhu, Yongqiu; Goverde, Rob M.P.

Publication date:
2020-10

Permanent link:
https://doi.org/10.3929/ethz-b-000486717

Originally published in:
Transportation Research Part B: Methodological 140, https://doi.org/10.1016/j.trb.2020.09.001

Transportation Research Part B 140 (2020) 282–314

Contents lists available at ScienceDirect

Transportation Research Part B

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

Integrated timetable rescheduling and passenger reassignment during railway disruptions

Yongqiu Zhu*, Rob M.P. Goverde

Department of Transport and Planning, Delft University of Technology, The Netherlands

A R T I C L E I N F O

Article history:
Received 17 July 2019
Revised 18 August 2020
Accepted 1 September 2020
Available online 9 September 2020

Keywords:
Railways
Disruption management
Timetable rescheduling
Passenger reassignment
Fix-and-optimize algorithm

A B S T R A C T

During railway disruptions, most passengers may not be able to find preferred alternative train services due to the current way of handling disruptions that does not take passenger responses into account. To offer better alternatives to passengers, this paper proposes a novel passenger-oriented timetable rescheduling model, which integrates timetable rescheduling and passenger reassignment into a Mixed Integer Linear Programming model with the objective of minimizing generalized travel times: in-vehicle times, waiting times at origin/transfer stations and the number of transfers. The model applies the dispatching measures of re-timing, re-ordering, cancelling, flexible stopping and flexible short-turning trains, handles rolling stock circulations at both short-turning and terminal stations of trains, and takes station capacity into account. To solve the model efficiently, an Adapted Fix-and-Optimize (AFaO) algorithm is developed. Numerical experiments were carried out to a part of the Dutch railways. The results show that the proposed passenger-oriented timetable rescheduling model is able to shorten generalized travel times significantly compared to an operator-oriented timetable rescheduling model that does not consider passenger responses. By allowing only 10 min more train delay than an optimal operator-oriented rescheduling solution, the passenger-oriented model is able to shorten the generalized travel times over all passengers by thousands of minutes in all considered disruption scenarios. With a passenger-oriented rescheduled timetable, more passengers continue their train travels after a disruption started, compared to a rescheduled timetable from the operator-oriented model. The AFaO algorithm obtains high-quality solutions to the passenger-oriented model in up to 300 s.

1. Introduction

Railway systems play an important role in people's daily travelling so that the operations are required as reliable as possible to ensure passenger punctuality. Unfortunately, unexpected disruptions occur in the railways on a daily basis (Zhu and Goverde, 2017), during which many train services are delayed and cancelled that disturb passenger planned journeys significantly. When rescheduling a timetable in case of a disruption, traffic controllers decide which services have to be delayed or cancelled in terms of pre-designed contingency plans, where the impact on passengers is considered to a very limited extent

* Corresponding author.
E-mail addresses: y.zhu-5@tudelft.nl (Y. Zhu), r.m.p.goverde@tudelft.nl (R.M.P. Goverde).

https://doi.org/10.1016/j.trb.2020.09.001

(Ghaemi et al., 2017b). As a result, the rescheduled train services may not be passenger-friendly. For example, passengers may hardly find alternative train services to reach the expected destinations in reasonable travel times. To provide passengers with better alternatives during disruptions, it is necessary to reschedule a timetable in a more passenger-oriented way.

1.1. The state-of-the-art on timetable rescheduling

Passenger-oriented timetable rescheduling started from the field of *delay management* that decides whether a train should wait for a delayed feeder train to guarantee the transfer connection of some passengers. Schöbel (2001) is the first one dealing with this problem based on the assumption that if passengers missed the transfer connections, they would wait for a complete cycle time to catch the next connection considering that the planned timetable is periodic. Dollevoet et al. (2012) make an extension by introducing the possibility of rerouting passengers who are assumed to take the shortest paths for their following travels in case of missed transfers. Both papers describe the infrastructure at a macroscopic level neglecting signals and block sections. To improve solution feasibility in practice, Corman et al. (2017) propose a delay management model in which the infrastructure is described at a microscopic level. Albert et al. (2017) formulate passenger behaviours in stations (e.g. queueing in boarding trains) at a microscopic level to describe passenger influences on train delays rather than considering the impact of train delays on passenger behaviours only.

Delay management deals with the interaction between timetable and passengers, but not the interaction between timetable and reduced infrastructure availability, which however must be taken into account by disruption management. Operator-oriented disruption management considers only the latter kind of interaction, while passenger-oriented disruption management considers both kinds of interactions. In practice, disruption management consists of three phases starting from the disruptive event (failure) (Ghaemi et al., 2017b). The first phase consists of getting information about the disruption and its location, guaranteeing safety, estimating the expected duration and deciding on the rescheduling measures. In the second phase the rescheduled timetable is applied and in the third phase the traffic recovers to the original timetable. At present, the first phase can take up quite some time depending on how existing contingency plans need to be adjusted, how many changes have to be made to the dispatching plans, and how drivers can be informed of disruptions ahead. Speeding up this process is required to avoid queuing of stranded trains. A time limit of 5 minute to compute a rescheduling solution will be sensible for mainline railway networks and would imply a big improvement on the current practice. Note that this paper handles serious disruptions of blocked tracks that go beyond simple re-timing or re-ordering decisions.

Most literature on disruption management is operator-oriented, including (Ghaemi et al., 2017, 2018; Meng and Zhou, 2011; Veelenturf et al., 2015; Zhan et al., 2015, 2016; Zhu and Goverde, 2019). The differences among these papers lie in the considered railway lines (single-track lines or double-track lines), the adopted dispatching measures, whether considering the transition from the planned timetable to the disruption timetable and vice versa, the extent of infrastructure description (macroscopic or microscopic level), the number of considered disruptions (single disruption or multiple disruptions), and/or the characteristic of disruption length (deterministic or uncertain). The similarity among these papers is that they all use operator-oriented objectives: e.g., minimizing train delays and/or cancellations, in which a constant cancellation penalty is used to represent the delay of cancelling each train. There are a few papers that consider both operators and passengers. Bettinelli et al. (2017) associate dispatching decisions with different penalties considering the extents of their impacts on passengers. For example, a major change in a train path is associated with a bigger penality. Louwerse and Huisman (2014) include a term in the objective to balance the numbers of cancelled trains in both directions to distribute the disruption impact evenly over the different passenger groups in case of partial track blockage.

A few works focus on passenger-oriented disruption management. Cadarso et al. (2013) propose a two-step approach in which a frequency-based passenger assignment model is performed first to estimate the passenger demand and then a rescheduling model (for timetable and rolling stock) is solved to accommodate the passenger demand as much as possible. The adopted dispatching measures are limited to cancelling original trains and inserting additional trains. Zhu and Goverde (2019c) adopt a schedule-based passenger assignment model to obtain the travel path of each passenger in terms of the planned timetable. With this information, the potential impact of each dispatching decision on passenger planned travels is estimated, which is used as weight in the objective to minimizing passenger delays. The adopted dispatching measures include re-timing, re-ordering, cancelling, flexible stopping (i.e. adding extra stops and skipping scheduled stops), and flexible short-turning. Short-turning a train means that a train stops at the last possible station before the blocked tracks and the corresponding rolling stock turns at that station to serve the opposite operation. *Flexible* short-turning means that each train is given a full choice of short-turning station candidates, and the model decides the optimal station and time of short-turning a train. Both Cadarso et al. (2013) and Zhu and Goverde (2019c) consider static passenger demand, which neglect that passengers may choose other travel paths rather than the planned ones due to the rescheduled train services. To formulate passenger behaviour in a more realistic way, it is necessary to take into account passenger responses towards the rescheduled train services. Veelenturf et al. (2017) propose an iterative approach that embeds a timetable rescheduling model and a passenger assignment model into an iterative framework where at each iteration an adjustment will be applied on the timetable if it reduces the total passenger inconvenience as evaluated by the passenger assignment model. The adjustments are restricted to adding stops. Binder et al. (2017) propose an integrated approach of formulating the timetable rescheduling and the passenger assignment into one single model that computes a rescheduled timetable by an optimization solver directly. The applied dispatching measures include re-timing, re-ordering, cancelling, global re-routing and inserting additional trains. The rolling stock circulations at the short-turning and terminal stations of trains are neglected. Gao et al. (2016) also propose a timetable rescheduling model considering dynamic passenger flows, while focusing on the recovery phase of a disruption. As the target case is a metro corridor, all passengers are assumed to choose direct trains (i.e. no transfers). The dispatching measures of stop-skipping and re-timing are used to adjust the timetable to reduce passenger waiting times at stations. Due to the computational complexity, the master problem of generating a rescheduled timetable is decomposed into a series of sub-problems that each reschedules one train only. When solving a sub-problem for one train, the stopping patterns and time schedules of the previous considered trains are all fixed.

1.2. The scientific gaps on passenger-oriented timetable rescheduling

Formulating passenger re-routing as a multi-commodity flow problem is a method commonly used in the literature. For example in Binder et al. (2017) and Corman et al. (2017), a timetable is formulated into a directed acyclic graph (DAG) to describe passenger path choices. Then, the passenger re-routing is modelled as a multi-commodity flow problem, in which passengers flow through the arcs of the DAG that is updated according to the rescheduled timetable. The challenges of modelling passenger re-routing this way mainly lie in two aspects: (1) how to *formulate* a DAG from a timetable to describe more path attributes with limited nodes/arcs, and (2) how to *reformulate* a DAG dynamically during timetable rescheduling when passenger re-routing is integrated. The existing literature either uses a simple method of formulating a DAG, which cannot reflect certain path attributes (e.g. the number of transfers), or adopts a formulation method that will lead to a large-size of DAG if focusing on a railway network with high-frequency services. Also, limited dispatching measures (e.g. no flexible stopping) are used in the literature, which need to be extended to explore more alternative path choices for passengers during disruptions. However, including more dispatching measures will increase the complexity of reformulating a DAG during timetable rescheduling. Another challenge is designing an efficient algorithm to solve the integrated timetable rescheduling and passenger re-routing model with high-quality solutions in an acceptable time. This has been reported as a challenging task in the literature so far (Corman et al., 2017; Binder et al., 2017).

1.3. The contributions of this paper

This paper contributes to the literature by improved methods of DAG formulation and reformulation to enable a better integrated timetable rescheduling and passenger re-routing model in terms of the considerations of multiple path attributes and multiple dispatching measures. This paper also contributes with an efficient algorithm to solve the integrated model with optimal or near-optimal solutions. The key contributions of this paper are summarized as follows.

- An improved method of formulating a DAG (called an event-activity network in this paper) from a timetable is proposed, by explicitly distinguishing passenger activities at origin stations, transfer stations (if any) and trains without time discretization.
- A new concept, the transition network, is proposed to enable the dynamic formulation of event-activity networks considering the impacts of multiple dispatching measures, the characteristics of the disruption, the operational requirements of trains, and the travel requirements of passengers.
- For the first time, the dispatching measure of flexible stopping (adding and skipping stops) is formulated with passenger re-routing in a railway network (instead of one corridor) where transfers are allowed.
- An adapted fix-and-optimize (AFaO) algorithm is designed to iteratively solve the proposed passenger-oriented timetable rescheduling model. The algorithm allows to balance the solution quality and computation time by changing the input parameter.
- The passenger-oriented timetable rescheduling model is able to generate rescheduling solutions with shorter generalized travel times than an operator-oriented model according to results of real-life instances in part of the Dutch railway network.

This paper considers single disruption that blocks tracks between stations completely assuming that the duration of the disruption is known at the beginning of the disruption, and will not change over time. We describe infrastructure at a macroscopic level and handle railway networks with both single-track and double-track railway lines. We use the dispatching measures of re-timing, re-ordering, cancelling, flexible short-turning, and flexible stopping to compute a feasible rescheduled timetable from the start of a disruption until it is fully recovered. A train is assumed to have unlimited capacity, which means that a passenger is able to board any train if he/she decides to board this train. This is because we focus on providing better alternative train services to passengers so that the possible impact of vehicle capacity on passengers is neglected. In this way, we can get the optimal rescheduled timetable in terms of generalized travel times. This optimal rescheduled timetable can then be used as an input to rolling stock rescheduling that aims to accommodate the passenger demand as much as possible. For example, Kroon et al. (2014) and Van der Hurk et al. (2018) both deal with passenger-oriented rolling stock rescheduling with a rescheduled timetable given as input.

The remainder of the paper is organized as follows. Section 2 introduces the general framework of establishing the passenger-oriented timetable rescheduling model. Section 3 explains how to formulate a timetable into an event-activity network, which is a directed acyclic graph with events as nodes and activities as arcs to describe passenger path choices. A

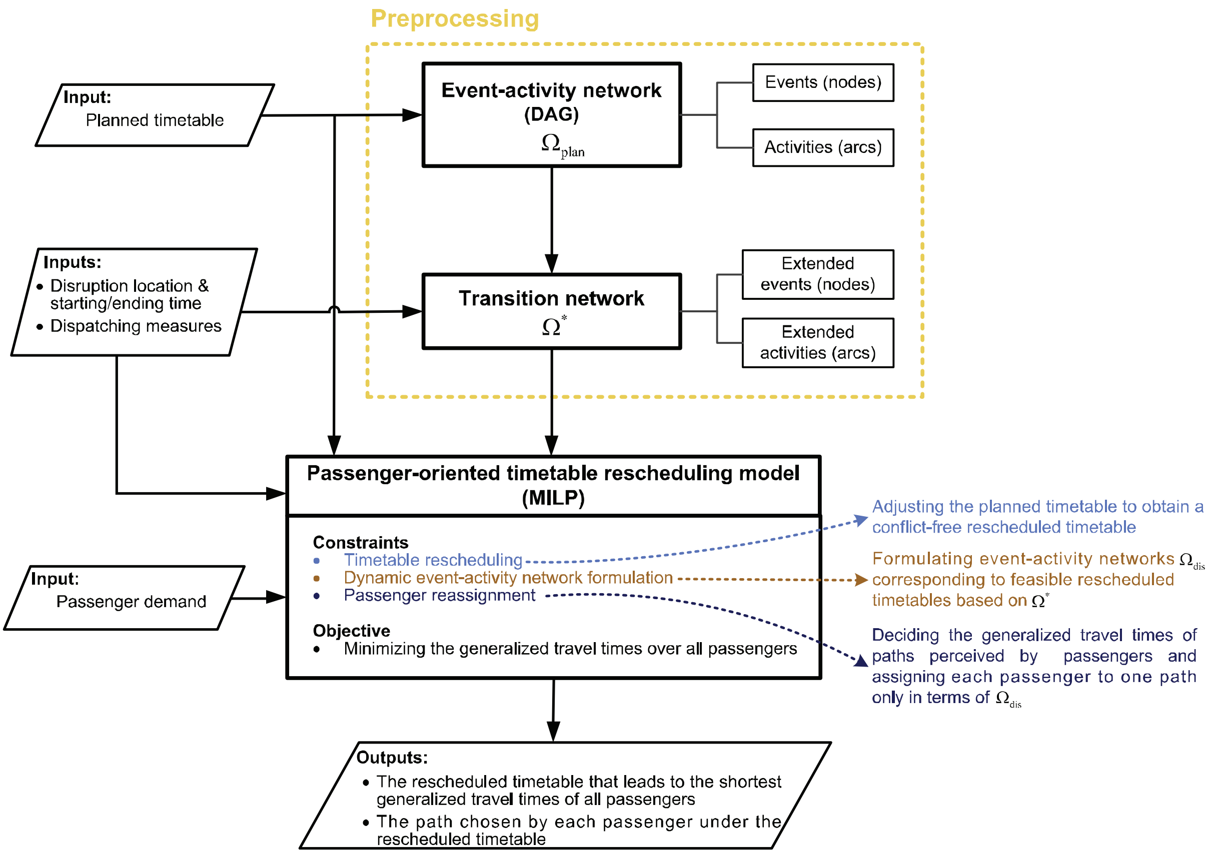


Fig. 1. An overview of the passenger-oriented timetable rescheduling model.

path is constituted by a series of connected events and activities. The planned timetable can be formulated into an event-activity network Ω_{plan}, which is then extended to a transition network Ω^* that enables the dynamic formulation of event-activity networks during timetable rescheduling. A transition network is a combination of all events and activities that could be in any event-activity networks formulated from feasible rescheduled timetables towards the disruption concerned. Section 4 introduces the method of constructing a transition network. Based on a transition network, the passenger-oriented timetable rescheduling model is proposed in Section 5 followed by Section 6 that introduces the methods of reducing the computational complexity of the model. In Section 7, extensive numerical experiments were carried out to a part of the Dutch railways. Finally, Section 8 concludes the paper and points out future research directions.

2. General framework

This paper integrates timetable rescheduling with passenger re-routing into an MILP model, for which two preprocessing steps are needed. Fig. 1 gives an overview of the model.

The first preprocessing step transforms the planned timetable into an event-activity network Ω_{plan}, which is a directed acyclic graph used to describe passenger path choices. The method of constructing an event-activity network from a timetable is introduced in Section 3. In case of a disruption, the planned timetable will become infeasible, and so does the corresponding event-activity network Ω_{plan} that now is unable to reflect the paths currently available in the railways. Under this circumstance, the timetable has to be rescheduled, and during rescheduling the corresponding event-activity networks have to be updated as well to consider timetable-dependent passenger behaviours. To enable a dynamic event-activity network formulation during timetable rescheduling, we perform the second preprocessing step to construct a transition network Ω^*. A transition network is extended from the event-activity network Ω_{plan} by adding all events and activities that could exist in any event-activity network Ω_{dis} corresponding to a feasible rescheduled timetable obtained for a specific disruption. In other words, $\Omega^* = \bigcup_i \Omega^i_{dis} \cup \Omega_{plan}$, where Ω^i_{dis} refers to the event-activity network corresponding to the ith feasible rescheduled timetable. For one specific disruption there are usually multiple feasible rescheduled timetables. Note that Ω^* varies with the disruption characteristics (i.e. location and starting/ending time) and the dispatching measures allowed. A transition network Ω^* is *not* a directed acyclic graph as it includes the possibility of changing the order of trains. The method of constructing a transition network is introduced in Section 4.

Table 1
Event attributes.

Symbol	Description
st_e	The corresponding station of event $e \in E \backslash E_{\text{penal}}$
tr_e	The corresponding train of event $e \in E_{\text{ar}} \cup E_{\text{de}} \cup E_{\text{dde}}$
tl_e	The corresponding train line of event $e \in E_{\text{ar}} \cup E_{\text{de}} \cup E_{\text{dde}}$
λ_e	The corresponding departure event of $e \in E_{\text{dde}}$
o_e	The scheduled time of event $e \in E_{\text{ar}} \cup E_{\text{de}} \cup E_{\text{dde}}$

The constructed transition network, the planned timetable, the disruption characteristics, and the allowed dispatching measures are all necessary inputs to establish the passenger-oriented timetable rescheduling model, which is formulated as an MILP in this paper. This model consists of the constraints for three purposes: 1) timetable rescheduling, 2) dynamic event-activity network formulation, and 3) passenger reassignment. The timetable rescheduling constraints ensure a rescheduled timetable does not violate any infrastructure and operational restrictions. The constraints relevant to the dynamic event-activity network formulation decide which activities and events of Ω^* should be selected to construct an event-activity network Ω_{dis} in terms of a rescheduled timetable. The passenger reassignment constraints decide the weight of each activity of Ω_{dis} from the perspectives of passengers, and assign each passenger to one path only. A path is described by a sequence of connected activities. The total activity weight of a path is the generalized travel time of this path. The objective of the model is minimizing the generalized travel times of all passengers. By this model, a rescheduled timetable that leads to the shortest generalized travel times of all passengers can be obtained, as well as the path chosen by each passenger under the rescheduled timetable.

3. Event-activity network

This section defines an event-activity network, which is a representation of a timetable and allows passenger path choices to be described. An event-activity network needs to be reconstructed if the corresponding timetable is rescheduled. This section introduces how to formulate an event-activity network given a fixed timetable.

3.1. Events

Six types of events are created in an event-activity network. They are arrival events, departure events, duplicate departure events, entry events, exit events and a penalty event, which constitute the sets E_{ar}, E_{de}, E_{dde}, E_{entry}, E_{exit} and E_{penal}, respectively. In particular, $E_{\text{ar}} = E_{\text{ar}}^{\text{alight}} \cup E_{\text{ar}}^{\text{pass}}$, and $E_{\text{de}} = E_{\text{de}}^{\text{board}} \cup E_{\text{de}}^{\text{pass}}$, where $E_{\text{ar}}^{\text{alight}}$ is the set of arrival events that correspond to passenger alighting, and $E_{\text{de}}^{\text{board}}$ is the set of departure events that correspond to passenger boarding. The arrival (departure) events associated to a through train that do *not* correspond to passenger alighting (boarding) constitute the set of $E_{\text{ar}}^{\text{pass}}$ ($E_{\text{de}}^{\text{pass}}$).

The attributes of events are indicated in Table 1. Note that an event $e \in E_{\text{dde}}$ is the duplicate of a departure event $e' \in E_{\text{de}}^{\text{board}}$ with exactly the same attributes which e' has, and with an extra attribute λ_e to indicate the departure event e' corresponding to e: $E_{\text{dde}} = \{e | \lambda_e = e', e' \in E_{\text{de}}^{\text{board}}\}$. One and only one duplicate is created for a departure event $e' \in E_{\text{de}}^{\text{board}}$. Duplicate departure events are used for constructing wait, boarding and transfer activities, which are explained in more detail in Section 3.2. Note that this paper defines these activities differently than Zhu and Goverde (2019a). As for E_{penal}, it contains only one penalty event for constructing the penalty arcs that enable passengers who cannot find preferred paths to leave the railways.

3.2. Activities

An activity is a directed arc between two different events. Ten types of activities are created in an event-activity network, which are constructed as follows.

$A_{\text{entry}} = \{(e, e') | e \in E_{\text{entry}}, e' \in E_{\text{dde}}, st_e = st_{e'}\}$. *Entry activities* enable passengers to enter the railways when arriving at the origins.

$A_{\text{exit}} = \{(e, e') | e \in E_{\text{ar}}^{\text{alight}}, e' \in E_{\text{exit}}, st_e = st_{e'}\}$. *Exit activities* enable passengers to leave the railways when arriving at the destinations,

$A_{\text{enpenal}} = \left\{\left(e, e'\right) \middle| e \in E_{\text{entry}}, e' \in E_{\text{penal}}\right\}$, and $A_{\text{expenal}} = \left\{\left(e'e\right) \middle| e' \in E_{\text{penal}}, e \in E_{\text{exit}}\right\}$. *Entry penalty activities* and *exit penalty activities* together enable passengers to drop the railways in case no preferred paths can be found,

$A_{\text{board}} = \left\{\left(e, e'\right) \middle| e \in E_{\text{dde}}, e' \in E_{\text{de}}^{\text{board}}, e' = \lambda_e\right\}$. *Boarding activities* enable passengers to board a train. Each duplicate departure event is linked to its corresponding departure event.

$A_{\text{run}} = \left\{\left(e, e'\right) \middle| e \in E_{\text{de}}, e' \in E_{\text{ar}}, tr_e = tr_{e'}, st_e \text{ is the upstream station adjacent to } st_{e'}\right\}$. *Running activities* enable passengers to travel from one station to another in a train.

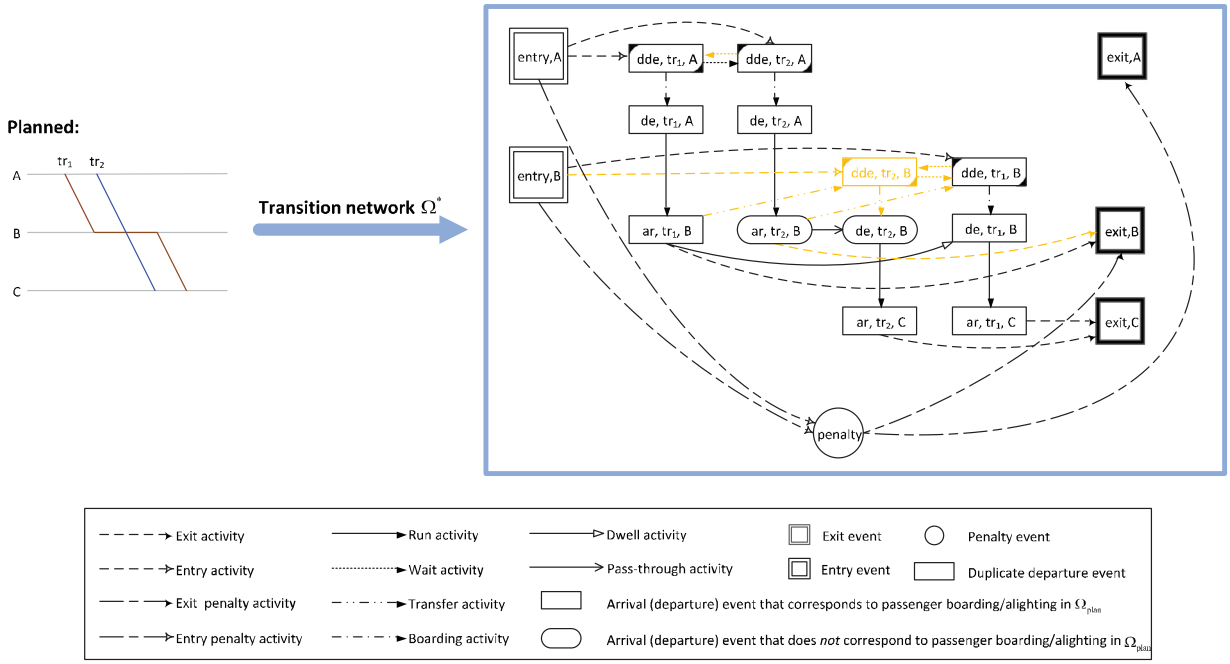


Fig. 2. A planned timetable with the constructed transition network.

$A_{\text{dwell}} = \left\{ \left(e, e'\right) \middle| e \in E_{\text{ar}}^{\text{alight}}, e' \in E_{\text{de}}^{\text{board}}, tr_e = tr_{e'}, st_e = st_{e'}, o_{e'} - o_e > 0 \right\}$. *Dwell activities* enable passengers to wait at a station in a train.

$A_{\text{pass}} = \left\{ \left(e, e'\right) \middle| e \in E_{\text{ar}}^{\text{pass}}, e' \in E_{\text{de}}^{\text{pass}}, tr_e = tr_{e'}, st_e = st_{e'}, o_{e'} - o_e = 0 \right\}$. *Pass-through activities* enable passengers to pass through a station in a train. Note that it is necessary to distinguish the planned pass-through and dwell activities so that we can recognize the skipped(extra) stops in a rescheduled timetable because the dispatching measure of flexible stopping is applied in this paper.

$A_{\text{wait}} = \left\{ \left(e, e'\right) \middle| e \in E_{\text{dde}}, e' = \arg\min \left\{ o_{e'} | o_{e'} \geq o_e, e' \in E_{\text{dde}}, tr_{e'} \neq tr_e, st_{e'} = st_e \right\} \right\}$. *Wait activities* enable passengers to wait at a station. Each duplicate departure event is linked to the next time-closest duplicate departure event that is at the same station but corresponds to another train.

$A_{\text{trans}} = \left\{ \left(e, e'\right) \middle| e \in E_{\text{ar}}^{\text{alight}}, e' = \arg\min \left\{ o_{e'} | o_{e'} \geq o_e + \ell_{e,e'}^{\text{trans}}, e' \in E_{\text{dde}}, tr_{e'} \neq tr_e, st_{e'} = st_e \right\} \right\}$. *Transfer activities* enable passengers to transfer from one train to another. Each arrival event is linked to the next time-closest duplicate departure event that occurs at least $\ell_{e,e'}^{\text{trans}}$ later at the same station but corresponds to another train. Here, $\ell_{e,e'}^{\text{trans}}$ represents the minimum transfer time required from the arrival train tr_e to another departure train $tr_{e'}$, which are alongside the same platform or different platforms affecting the value of $\ell_{e,e'}^{\text{trans}}$.

An event-activity network is $\Omega = (E, A)$, which is a directed acyclic graph (DAG). In the blue box of Fig. 2, all nodes and arcs colored in black constitute an event-activity network formulated from the planned timetable shown in the left.

3.3. Weights of activities

The weights of activities are determined as follows:

$$
\begin{array}{ll}
w_a = \beta_{\text{vehicle}}(o_{e'} - o_e), & a = (e, e') \in A_{\text{run}} \cup A_{\text{dwell}} \cup A_{\text{pass}}, \\
w_a = \beta_{\text{wait}}(o_{e'} - o_e), & a = (e, e') \in A_{\text{wait}}, \\
w_a = \beta_{\text{wait}}(o_{e'} - o_e) + \beta_{\text{trans}}, & a = (e, e') \in A_{\text{trans}}, \\
w_a^g = o_{e'} - t_g^{\text{ori}}, & a = (e, e') \in A_{\text{entry}} : o_{e'} \geq t_g^{\text{ori}} \\
w_a^g = T_g^{\text{max}}, & a = (e, e') \in A_{\text{enpenal}} \\
w_a = 0, & a = (e, e') \in A_{\text{board}} \cup A_{\text{exit}} \cup A_{\text{expenal}},
\end{array}
$$

where β_{vehicle} and β_{wait} represent respectively passenger preference on in-vehicle times and waiting times at stations, and β_{trans} refers to the time penalty of one transfer. Note that the weight of an entry activity or entry penality activity is passenger dependent. t_g^{ori} is the time when passenger group g arrives at the origin station, and this paper assumes that each passenger group g has an acceptable maximum generalized travel time T_g^{max} in the railways.

Table 2
Sets relevant to a transition/event-activity network.

Notation	Description
Ω^*	Transition network: $\Omega^* = (E^*, A^*)$
Ω_{plan}	Event-activity network formulated from the planned timetable: $\Omega_{\text{plan}} = \left(E^{\text{plan}}, A^{\text{plan}}\right)$ and $\Omega_{\text{plan}} \subset \Omega^*$
Ω_{dis}	Event-activity network formulated from any possible disruption timetable by adjusting the planned timetable: $\Omega_{\text{dis}} \subset \Omega^*$
E^*	Set of events in Ω^*
E^{plan}	Set of events in Ω_{plan}: $E^{\text{plan}} \subset E^*$
E_i^{plan}	Set of i events in Ω_{plan}, $i \in$ {ar, de, dde, entry, exit, penal}: $E_i^{\text{plan}} \subset E^{\text{plan}}$
$E_{\text{ar}}^{\text{alight,plan}}$	Set of arrival events that correspond to passenger alighting in Ω_{plan}: $E_{\text{ar}}^{\text{alight,plan}} \subseteq E_{\text{ar}}^{\text{plan}}$
$E_{\text{ar}}^{\text{pass,plan}}$	Set of arrival events that do *not* correspond to passenger alighting in Ω_{plan}: $E_{\text{ar}}^{\text{pass,plan}} = E_{\text{ar}}^{\text{plan}} \backslash E_{\text{ar}}^{\text{alight,plan}}$
$E_{\text{de}}^{\text{board,plan}}$	Set of departure events that correspond to passenger boarding in Ω_{plan}: $E_{\text{de}}^{\text{board,plan}} \subseteq E_{\text{de}}^{\text{plan}}$
$E_{\text{de}}^{\text{pass,plan}}$	Set of departure events that do *not* correspond to passenger boarding in Ω_{plan}: $E_{\text{de}}^{\text{pass,plan}} = E_{\text{de}}^{\text{plan}} \backslash E_{\text{de}}^{\text{board,plan}}$
A^*	Set of activities in Ω^*
A_i^*	Set of i activities in Ω^*: $A_i^* \subset A^*$, $i \in$ {wait, trans, board, entry, exit}
A^{plan}	Set of activities in Ω_{plan}: $A^{\text{plan}} \subset A^*$
A_i^{plan}	Set of i activities in Ω_{plan}: $A_i^{\text{plan}} \subset A^{\text{plan}}$, $i \in$ {run, dwell, pass, wait, trans, board, entry, exit, enpenal, expenal}
A_k^{undis}	Set of undisrupted k activities in Ω^*: $A_k^{\text{undis}} \subset A_k^{\text{plan}}$, $k \in$ {run, dwell, pass, wait, trans, board, entry, exit}
$A_{k_1}^{\text{dis}}$	Set of disrupted k_1 activities in Ω^*: $A_{k_1}^{\text{dis}} = A_{k_1}^{\text{plan}} \backslash A_{k_1}^{\text{undis}}$, $k_1 \in$ {run, dwell, pass}
$A_{k_2}^{\text{dis}}$	Set of disrupted k_2 activities in Ω^*: $A_{k_2}^{\text{dis}} = A_{k_2}^* \backslash A_{k_2}^{\text{undis}}$, $k_2 \in$ {wait, trans, board, entry, exit}

4. Transition network

This section defines a transition network, which allows a dynamic event-activity network formulation during timetable rescheduling. The transition network Ω^* is an extension of the event-activity network Ω_{plan} formulated from the planned timetable by adding all events and activities that could exist in any rescheduled timetables. In other words, Ω^* represents all possible timetable adjustments, which can be used to describe the alternative paths available to passengers during timetable rescheduling. Before giving the details of constructing a transition network, an example on a simple case is given below to explain the basic idea.

Example Fig. 2 shows a planned timetable with three stations A, B and C, and two trains tr_1 and tr_2. Both trains start from A and end at B with train tr_1 additionally stopping at B. In the blue box, the events and activities in black constitute the event-activity network Ω_{plan} from the planned timetable, while the events and activities in black and orange together constitute the transition network Ω^*. In this case, Ω^* is extended from Ω_{plan} by adding a new event and eight new activities (colored in orange) that do not exist in the planned timetable but could exist in a rescheduled timetable. Due to the dispatching measure of re-ordering, train tr_1 could depart later than train tr_2 at station A, although train tr_1 was originally planned to depart earlier than train tr_2. Considering this possible train order change, an extra wait activity is added from event (dde, tr_2, A) to event (dde, tr_1, A), which creates a cycle between both events. This *disables* a transition network to be a DAG. Due to the dispatching measure of flexible stopping, an extra stop could be added to train tr_2 at station B. Thus, a new event (dde, tr_2, B) is added as well as an entry activity, a boarding activity, a wait activity, two transfer activities, and an exit activity. As can be seen entry/exit penalty activities always remain the same in Ω^* as in Ω_{plan}.

In the following, we introduce how to construct a transition network by extending the event-activity network Ω_{plan} corresponding to a planned timetable. The set notation with the superscript of *plan* represents the events/activities sets in Ω_{plan}. The set notation with the superscript of * represents the extended events/activities in Ω^*. Table 2 shows the notation of sets relevant to a transition/event-activity network.

4.1. Extended events

All events of event-activity network Ω_{plan} are included in the transition network Ω^*, in which only the set of duplicate departure events is extended

$E_{\text{dde}}^* = \left\{ e \middle| \lambda_e = e', e' \in E_{\text{de}}^{\text{plan}} \right\}$, where $E_{\text{de}}^{\text{plan}} = E_{\text{de}}^{\text{board,plan}} \cup E_{\text{de}}^{\text{pass,plan}}$. Here, $E_{\text{de}}^{\text{board,plan}}$ and $E_{\text{de}}^{\text{pass,plan}}$ represent respectively the set of departure events that correspond and do *not* correspond to passenger boarding in the planned timetable. Recall that in an event-activity network, duplicates are only created for departure events that correspond to passenger boarding.

4.2. Extended activities

All activities of event-activity network Ω_{plan} are included in the transition network Ω^*, in which five types of activities are extended including $A_{\text{wait}}^*, A_{\text{trans}}^*, A_{\text{board}}^*, A_{\text{entry}}^*$ and A_{exit}^*. Except entry/exit penalty activities, each type of activities is classified into two subsets, *undisrupted* and *disrupted*:

$$A_i^{\text{plan}} = A_i^{\text{undis}} \cup A_i^{\text{dis}}, \qquad i \in \{\text{run}, \text{dwell}, \text{pass}\},$$
$$A_k^* = A_k^{\text{undis}} \cup A_k^{\text{dis}}, \qquad k \in \{\text{wait}, \text{trans}, \text{board}, \text{entry}, \text{exit}\}.$$

We define an activity an *undisrupted* activity if both of the two events in this activity were originally planned to occur before t_{start} or after $t_{\text{end}} + R$, in which R is the time length required for the normal schedule to be fully recovered after the disruption ends. In that sense, an undisrupted activity is an activity that will *never* be different than planned in the rescheduled timetable. In this paper, we ensure an arrival (departure) event that was originally scheduled to occur before the disruption start t_{start} or at least R minutes later than the disruption end t_{end} will not be delayed/cancelled. This also applies to duplicate departure events, which are always with the same occurrence times as their corresponding departure events. We define an activity a *disrupted* activity if at least one of the two events in this activity could be cancelled or delayed. In that sense, a disrupted activity is an activity that could be different than planned in the rescheduled timetable. This paper requires that only the events, which were originally planned to occur during the period $[t_{\text{start}}, t_{\text{end}} + R]$ could be cancelled or delayed. These events can correspond to any stations, which are not distinguished between disrupted and undisrupted in the paper. Based on these, we decide whether an activity is undisrupted or disrupted as follows.

4.2.1. Running, dwell, and pass-through activities

The disrupted and undisrupted running, dwell, and pass-through activities are respectively defined as

$$A_i^{\text{dis}} = \left\{ \left(e, e'\right) \in A_i^{\text{plan}} | t_{\text{start}} \le o_e < t_{\text{end}} + R \text{ or } t_{\text{start}} \le o_{e'} < t_{\text{end}} + R \right\}, \qquad i \in \{\text{run}, \text{dwell}, \text{pass}\},$$

$$A_i^{\text{undis}} = A_i^{\text{plan}} \backslash A_i^{\text{dis}}, \, i \in \{\text{run}, \text{dwell}, \text{pass}\},$$

where o_e refers to the original scheduled time of e, t_{start} (t_{end}) represents the start (end) time of the disruption, and R represents the duration required for the disruption timetable resuming to the planned timetable after the disruption ends.

4.2.2. Entry, exit, and boarding activities

The disrupted entry activities are defined as $A_{\text{entry}}^{\text{dis}} = A_{\text{entry}}^{\text{dis},1} \cup A_{\text{entry}}^{\text{dis},2}$, where

$$A_{\text{entry}}^{\text{dis},1} = \left\{ \left(e, e'\right) \in A_{\text{entry}}^{\text{plan}} | t_{\text{start}} \le o_{e'} < t_{\text{end}} + R \right\},$$

$$A_{\text{entry}}^{\text{dis},2} = \left\{ \left(e, e'\right) \middle| e \in E_{\text{entry}}^{\text{plan}}, e' \in E_{\text{dde}}^{*} \backslash E_{\text{dde}}^{\text{plan}}, st_e = st_{e'}, t_{\text{start}} \le o_{e'} < t_{\text{end}} + R \right\}.$$

The disrupted exit activities are defined as $A_{\text{exit}}^{\text{dis}} = A_{\text{exit}}^{\text{dis},1} \cup A_{\text{exit}}^{\text{dis},2}$, where

$$A_{\text{exit}}^{\text{dis},1} = \left\{ \left(e, e'\right) \in A_{\text{exit}}^{\text{plan}} | t_{\text{start}} \le o_e < t_{\text{end}} + R \right\},$$

$$A_{\text{exit}}^{\text{dis},2} = \left\{ \left(e, e'\right) \middle| e \in E_{\text{ar}}^{\text{pass,plan}}, e' \in E_{\text{exit}}^{\text{plan}}, st_e = st_{e'}, t_{\text{start}} \le o_e < t_{\text{end}} + R \right\}.$$

The disrupted boarding activities are defined as $A_{\text{board}}^{\text{dis}} = A_{\text{board}}^{\text{dis},1} \cup A_{\text{board}}^{\text{dis},2}$, where

$$A_{\text{board}}^{\text{dis},1} = \left\{ \left(e, e'\right) \in A_{\text{board}}^{\text{plan}} | t_{\text{start}} \le o_{e'} < t_{\text{end}} + R \right\},$$

$$A_{\text{board}}^{\text{dis},2} = \left\{ \left(e, e'\right) \middle| e \in E_{\text{dde}}^{*} \backslash E_{\text{dde}}^{\text{plan}}, e' \in E_{\text{de}}^{\text{pass,plan}}, e' = \lambda_e, t_{\text{start}} \le o_{e'} < t_{\text{end}} + R \right\}.$$

$A_{\text{entry}}^{\text{dis},1}$, $A_{\text{exit}}^{\text{dis},1}$, and $A_{\text{board}}^{\text{dis},1}$ represent respectively the entry, exit, and boarding activities that could be cancelled due to the disruption. $A_{\text{entry}}^{\text{dis},2}$, $A_{\text{exit}}^{\text{dis},2}$, and $A_{\text{board}}^{\text{dis},2}$ represent respectively the entry, exit, and boarding activities that are not in Ω_{plan} but might be needed due to extra stops added in a rescheduled timetable. The undisrupted entry, exit, and boarding activities are respectively defined as $A_{\text{entry}}^{\text{undis}} = A_{\text{entry}}^{\text{plan}} \backslash A_{\text{entry}}^{\text{dis},1}$, $A_{\text{exit}}^{\text{undis}} = A_{\text{exit}}^{\text{plan}} \backslash A_{\text{exit}}^{\text{dis},1}$, and $A_{\text{board}}^{\text{undis}} = A_{\text{board}}^{\text{plan}} \backslash A_{\text{board}}^{\text{dis},1}$.

4.2.3. Wait activities

To construct disrupted wait activities, we first define three event sets, $E_{\text{dde}}^{\max} = \{\arg\max\{o_e | e \in E_{\text{dde}}^{\text{plan}}, o_e < t_{\text{start}}, st_e = st\}\}_{st \in ST}$, $E_{\text{dde}}^{\min} = \{\arg\min\{o_e | e \in E_{\text{dde}}^{\text{plan}}, o_e \ge t_{\text{end}} + R, st_e = st\}\}_{st \in ST}$, and $E_{\text{dde}}^{\text{dis}} = \{e \in E_{\text{dde}}^{*} | t_{\text{start}} \le o_e < t_{\text{end}} + R\}$, in which ST is the set of stations. Set $E_{\text{dde}}^{\max}$ includes at each station $st \in ST$ the latest duplicate departure event before t_{start}. Set $E_{\text{dde}}^{\min}$ includes at each station $st \in ST$ the earliest duplicate departure event after $t_{\text{end}} + R$. The events in $E_{\text{dde}}^{\max}$ and $E_{\text{dde}}^{\min}$ will *not* be affected by the disruption, while $E_{\text{dde}}^{\text{dis}}$ includes all duplicate departure events that could be affected by the disruption.

Based on $E_{\text{dde}}^{\max}$, $E_{\text{dde}}^{\min}$ and $E_{\text{dde}}^{\text{dis}}$, the set of disrupted wait activities is defined as $A_{\text{wait}}^{\text{dis}} = \bigcup_{j \in \{1,\ldots,4\}} A_{\text{wait}}^{\text{dis},j}$, in which

$$A_{\text{wait}}^{\text{dis},1} = \left\{ \left(e, e'\right) \middle| e \in E_{\text{dde}}^{\max}, e' \in E_{\text{dde}}^{\text{dis}}, st_{e'} = st_e, o_{e'} - o_e \le \ell_{\text{wait}}^{\max} \right\},$$

$$A_{\text{wait}}^{\text{dis},2} = \left\{ \left(e, e'\right) \middle| e \in E_{\text{dde}}^{\text{dis}}, e' \in E_{\text{dde}}^{\min}, st_{e'} = st_e, o_{e'} - o_e \le \ell_{\text{wait}}^{\max} + D \right\},$$

$$A_{\text{wait}}^{\text{dis},3} = \left\{ \left(e, e'\right) \middle| e, e' \in E_{\text{dde}}^{\text{dis}}, e \neq e', st_e = st_{e'}, 0 \le o_{e'} - o_e \le \ell_{\text{wait}}^{\max} + D \right\},$$

$$A_{\text{wait}}^{\text{dis},4} = \left\{ \left(e, e'\right) \middle| e, e' \in E_{\text{dde}}^{\text{dis}}, e \neq e', st_e = st_{e'}, -D \le o_{e'} - o_e < 0 \right\}.$$

Table 3
Decision variables.

Symbol	Description	Module
x_e	Continuous variable deciding the rescheduled time of an event $e \in E_{\text{ar}}^{\text{plan}} \cup E_{\text{de}}^{\text{plan}} \cup E_{\text{dde}}^{*}$.	1, 2, 3
c_e	Binary variable with value 1 deciding event $e \in E_{\text{ar}}^{\text{plan}} \cup E_{\text{de}}^{\text{plan}} \cup E_{\text{dde}}^{*}$ is cancelled, and 0 otherwise.	1, 2
s_a	Binary variable deciding whether a scheduled stop $a \in A_{\text{dwell}}^{\text{plan}}$ is skipped or whether an extra stop is added to $a \in A_{\text{pass}}^{\text{plan}}$. If $a \in A_{\text{dwell}}^{\text{plan}}$, then $s_a = 1$ indicates a is skipped. If $a \in A_{\text{pass}}^{\text{plan}}$, then $s_a = 1$ indicates a is added with a stop.	1, 2
y_a	Binary variable with value 1 deciding activity $a \in \Omega^*$ is effective in Ω_{dis}, and 0 otherwise.	2, 3
u_a^g	Binary variable with value 1 deciding activity $a \in \Omega^*$ is chosen by passenger group g, and 0 otherwise.	3
w_a^g	Continuous variable deciding the weight of activity $a \in \Omega^*$ perceived by each passenger in group g	3

Module 1: timetable rescheduling; Module 2: dynamic event-activity network formulation; Module 3: passenger reassignment

Here, D represents the maximum allowed delay per event, and $\ell_{\text{wait}}^{\max}$ represents the maximum waiting time that a passenger would like to spend at a station. We assume that $\ell_{\text{wait}}^{\max} \geq D$. Then, undisrupted wait activities are defined as $A_{\text{wait}}^{\text{undis}} = A_{\text{wait}}^{\text{plan}} \backslash (A_{\text{wait}}^{\text{plan}} \cap A_{\text{wait}}^{\text{dis}})$.

4.2.4. Transfer activities

To construct disrupted transfer activities, we first establish two event sets, $E_{\text{ar}}^{\text{dis}} = \{e | e \in E_{\text{ar}}^{\text{plan}}, t_{\text{start}} \leq o_e < t_{\text{end}} + R\}$, and $E_{\text{ar}}^{\text{trans}} = \{e | e \in E_{\text{ar}}^{\text{plan}}, o_e < t_{\text{start}}, (e, e') \in A_{\text{trans}}^{\text{plan}}, t_{\text{start}} \leq o_{e'} < t_{\text{end}} + R\}$. $E_{\text{ar}}^{\text{dis}}$ includes the arrival events that could be delayed/cancelled due to the disruption. $E_{\text{ar}}^{\text{trans}}$ contains the arrival events that will not be delayed/cancelled by the disruption but the corresponding planned transfer activities could be cancelled due to the disruption.

Based on $E_{\text{ar}}^{\text{dis}}$ and $E_{\text{ar}}^{\text{trans}}$, the disrupted transfer activities are defined as $A_{\text{trans}}^{\text{dis}} = \bigcup_{j \in \{1,\dots,5\}} A_{\text{trans}}^{\text{dis},j}$, where

$$A_{\text{trans}}^{\text{dis},1} = \left\{ \left(e, e'\right) \middle| e \in E_{\text{ar}}^{\text{trans}}, e' \in E_{\text{dde}}^{\min}, tr_{e'} \neq tr_e, st_{e'} = st_e, \ell_{e,e'}^{\text{trans}} \leq o_{e'} - o_e \leq \ell_{\text{trans}}^{\max} \right\},$$

$$A_{\text{trans}}^{\text{dis},2} = \left\{ \left(e, e'\right) \middle| e \in E_{\text{ar}}^{\text{trans}}, e' \in E_{\text{dde}}^{\text{dis}}, tr_{e'} \neq tr_e, st_{e'} = st_e, o_{e'} - o_e \leq \ell_{\text{trans}}^{\max} \right\},$$

$$A_{\text{trans}}^{\text{dis},3} = \left\{ \left(e, e'\right) \middle| e \in E_{\text{ar}}^{\text{dis}}, e' \in E_{\text{dde}}^{\min}, tr_{e'} \neq tr_e, st_{e'} = st_e, \ell_{e,e'}^{\text{trans}} \leq o_{e'} - o_e \leq \ell_{\text{trans}}^{\max} + D \right\},$$

$$A_{\text{trans}}^{\text{dis},4} = \left\{ \left(e, e'\right) \middle| e \in E_{\text{ar}}^{\text{dis}}, e' \in E_{\text{dde}}^{\text{dis}}, tr_{e'} \neq tr_e, st_{e'} = st_e, 0 \leq o_{e'} - o_e \leq \ell_{\text{trans}}^{\max} + D \right\},$$

$$A_{\text{trans}}^{\text{dis},5} = \left\{ \left(e, e'\right) \middle| e \in E_{\text{ar}}^{\text{dis}}, e' \in E_{\text{dde}}^{\text{dis}}, tr_{e'} \neq tr_e, st_{e'} = st_e, \ell_{e,e'}^{\text{trans}} - D \leq o_{e'} - o_e < 0 \right\},$$

Here, $\ell_{e,e'}^{\text{trans}}$ represents the minimum transfer time, and $\ell_{\text{trans}}^{\max}$ represents the maximum transfer time that a passenger would like to spend at a station. We assume that $\ell_{\text{trans}}^{\max} \geq D > \ell_{e,e'}^{\text{trans}}$. $A_{\text{trans}}^{\text{dis},1}$ and $A_{\text{trans}}^{\text{dis},2}$ are both related to $E_{\text{ar}}^{\text{trans}}$, while $A_{\text{trans}}^{\text{dis},3}$, $A_{\text{trans}}^{\text{dis},4}$ and $A_{\text{trans}}^{\text{dis},5}$ are all related to $E_{\text{ar}}^{\text{dis}}$. Undisrupted transfer activities are then defined as $A_{\text{trans}}^{\text{undis}} = A_{\text{trans}}^{\text{plan}} \backslash (A_{\text{trans}}^{\text{plan}} \cap A_{\text{trans}}^{\text{dis}})$.

5. Passenger-oriented timetable rescheduling model

In this section, we formulate the passenger-oriented timetable rescheduling problem as an MILP model, with the objective of minimizing generalized travel times. The MILP model consists of three constraint modules: 1) timetable rescheduling, 2) dynamic event-activity network formulation, and 3) passenger reassignment.

The timetable rescheduling module computes a feasible rescheduled timetable. The dynamic event-activity network formulation module formulates an event-activity network Ω_{dis} corresponding to the rescheduled timetable based on the preconstructed transition network Ω^*. The passenger reassignment module decides the weight of each activity $a \in \Omega^*$ perceived by each passenger, and assigns each passenger to the path with the shortest generalized travel time perceived by this passenger.

The constraints used in the timetable rescheduling module are all from Zhu and Goverde (2019c) so that we do not present them in this paper, neither the decision variables that are only used in this module. We refer to Zhu and Goverde (2019c) for details. In this paper, we present the constraints in the modules of the dynamic event-activity network formulation and the passenger reassignment, as well as the corresponding decision variables. Table 3 lists these decision variables and the modules in which they are used. The notation of parameters/sets can be found in Table 19 in the Appendix. Note that the rescheduled time x_e of any event e that was originally scheduled to occur before t_{start} or after $t_{\text{end}} + R$ is forced to be the same as its original scheduled time o_e by constraints from Zhu and Goverde (2019c). In other words, our model respects what has already happened before the beginning of the disruption, and recovers the disruption back to the normal schedule at latest R time after the end of the disruption.

Due to flexible stopping, scheduled stops could be skipped and extra stops could be added. The scheduled stops (non-stops) can also be cancelled, due to short-turning or complete train cancellation. Table show all possible stop types in a rescheduled timetable, and the corresponding values of the relevant decision variables. There are specific constraints in the timetable rescheduling module to limit the value combinations of $c_e, c_{e'}$ and s_a. We refer to Zhu and Goverde (2019c) for details.

5.1. Dynamic event-activity network formulation

The dynamic event-activity network formulation module decides which events and activities of the transition network Ω^* are effective in an event-activity network Ω_{dis} corresponding to a rescheduled timetable by respecting the rules of constructing an event-activity network introduced in Section 3. Recall that $\Omega^* = (E^*, A^*)$, where $E^* = E^{\text{plan}}_{\text{ar}} \cup E^{\text{plan}}_{\text{de}} \cup E^*_{\text{dde}} \cup E^{\text{plan}}_{\text{entry}} \cup E^{\text{plan}}_{\text{exit}} \cup E^{\text{plan}}_{\text{penal}}$, and $A^* = A^{\text{plan}}_{\text{run}} \cup A^{\text{plan}}_{\text{dwell}} \cup A^{\text{plan}}_{\text{pass}} \cup A^*_{\text{wait}} \cup A^*_{\text{trans}} \cup A^*_{\text{board}} \cup A^*_{\text{entry}} \cup A^*_{\text{exit}} \cup A^{\text{plan}}_{\text{enpenal}} \cup A^{\text{plan}}_{\text{expenal}}$. In particular, $A^{\text{plan}}_i = A^{\text{undis}}_i \cup A^{\text{dis}}_i, i \in \{\text{run}, \text{dwell}, \text{pass}\}$, and $A^*_j = A^{\text{undis}}_j \cup A^{\text{dis}}_j, j \in \{\text{wait}, \text{trans}, \text{board}, \text{entry}, \text{exit}\}$, which means that in the transition network Ω^*, each kind of activity set consists of two subsets: an undisrupted activity set, and a disrupted activity set. For an undisrupted activity, both of the corresponding events will not be delayed/cancelled by the disruption; while for a disruption activity, at least one of the corresponding events could be delayed/cancelled by the disruption.

5.1.1. Deciding which events are effective in Ω_{dis}

The binary cancellation decision c_e of an event $e \in E^{\text{plan}}_{\text{ar}} \cup E^{\text{plan}}_{\text{de}} \cup E^*_{\text{dde}}$ is equivalent to deciding whether this event is effective in Ω_{dis}. An event $e \in E^{\text{plan}}_{\text{ar}} \cup E^{\text{plan}}_{\text{de}} \cup E^*_{\text{dde}}$ is effective in Ω_{dis} if it is not cancelled, $c_e = 0$. The cancellation decision c_e and the rescheduled time x_e of an arrival (departure) event $e \in E^{\text{plan}}_{\text{ar}} (e \in E^{\text{plan}}_{\text{de}})$ are determined in the timetable rescheduling module. A duplicate departure event $e' \in E^*_{\text{dde}}$ is required to be cancelled/kept simultaneously as its corresponding departure event $e \in E^{\text{plan}}_{\text{de}}$, and the rescheduled times of both events are forced to be the same:

$$c_{e'} = c_e, \qquad e' \in E^*_{\text{dde}}, e \in E^{\text{plan}}_{\text{de}}, \lambda_{e'} = e, \tag{1}$$

$$x_{e'} = x_e, \qquad e' \in E^*_{\text{dde}}, e \in E^{\text{plan}}_{\text{de}}, \lambda_{e'} = e, \tag{2}$$

where $\lambda_{e'}$ is a given attribute indicating the departure event corresponding to duplicate departure event e'.

An event $e \in E^{\text{plan}}_{\text{entry}} \cup E^{\text{plan}}_{\text{exit}} \cup E^{\text{plan}}_{\text{penal}}$ is always effective in any Ω_{dis}.

5.1.2. Deciding which activities are always effective in any Ω_{dis}

Entry/exit penalty activities, and undisrupted activities are effective in any Ω_{dis}:

$$y_a = 1, \qquad a \in A^{\text{plan}}_{\text{enpenal}} \cup A^{\text{plan}}_{\text{expenal}}, \tag{3}$$

$$y_a = 1, \qquad a \in \left\{A^{\text{undis}}_k\right\}_{k\in K}, K = \{\text{run}, \text{dwell}, \text{pass}, \text{wait}, \text{trans}, \text{board}, \text{entry}, \text{exit}\}, \tag{4}$$

where y_a is a binary variable with value 1 indicating that activity a is effective in Ω_{dis}, and 0 otherwise. Recall that both of the events corresponding to an undisrupted activity will not be delayed/cancelled due to the disruption.

5.1.3. Deciding which disrupted run activities are effective in Ω_{dis}

Recall that a running activity is from a departure event e to an arrival event e', which correspond to the same train at neighbouring stations. A disrupted running activity in the transition network Ω^* will be effective in an event-activity network Ω_{dis} if neither of the corresponding events is cancelled:

$$y_a = 1 - c_e, \qquad a = (e, e') \in A^{\text{dis}}_{\text{run}}, \tag{5}$$

$$y_a = 1 - c_{e'}, \qquad a = (e, e') \in A^{\text{dis}}_{\text{run}}. \tag{6}$$

Note that in the timetable rescheduling module (Zhu and Goverde, 2019c), the departure event e and the arrival event e' in the same running activity are forced to be cancelled/kept simultaneously: $c_e = c_{e'}$, which is why we use equalities for (5) and (6).

5.1.4. Deciding which disrupted dwell/pass-through activities are effective in Ω_{dis}

Recall that a dwell (pass-through) activity is from an arrival event e to a departure event e', which correspond to the same train at the same station. We decide whether a disrupted dwell (pass-through) activity of Ω^* will be effective in Ω_{dis} by:

$$y_a \le 1 - c_e, \qquad a = (e, e') \in A^{\text{dis}}_{\text{dwell}} \cup A^{\text{dis}}_{\text{pass}}, \tag{7}$$

$$y_a \le 1 - c_{e'}, \qquad a = (e, e') \in A^{\text{dis}}_{\text{dwell}} \cup A^{\text{dis}}_{\text{pass}}, \tag{8}$$

$$y_a \ge 1 - c_e - c_{e'}, \qquad a = (e, e') \in A^{\text{dis}}_{\text{dwell}} \cup A^{\text{dis}}_{\text{pass}}. \tag{9}$$

Constraints (7) and (8) mean that a disrupted dwell (pass-through) activity will not be effective in Ω_{dis} if at least one of the corresponding events is cancelled; otherwise, it must be effective (9). Recall that $A^{\text{dis}}_{\text{dwell}} \subseteq A^{\text{plan}}_{\text{dwell}}$ and $A^{\text{dis}}_{\text{pass}} \subseteq A^{\text{plan}}_{\text{pass}}$.

Table 4
The stop type of activity $a = (e, e') \in A_{\text{dwell}}^{\text{plan}}$ in a rescheduled timetable according to $c_e, c_{e'}$ and s_a.

c_e	$c_{e'}$	s_a	Stop type
0	0	0	Stop
0	0	1	Skipped stop
1	0	0	Cancelled stop
0	1	0	Cancelled stop
1	1	0	Cancelled stop

Table 5
The stop type of activity $a = (e, e') \in A_{\text{pass}}^{\text{plan}}$ in a rescheduled timetable according to $c_e, c_{e'}$ and s_a.

c_e	$c_{e'}$	s_a	Stop type
0	0	0	Extra stop
0	0	1	Non-stop
1	0	1	Cancelled non-stop
0	1	1	Cancelled non-stop
1	1	1	Cancelled non-stop

5.1.5. Deciding which disrupted entry activities are effective in Ω_{dis}

Recall that an entry activity is from an entry event e to a duplicate departure event e', which both correspond to the same station. We use a binary parameter $r_{e'}$ with value 1 to indicate that a (duplicate) departure event e' corresponds to a train origin departure, and 0 otherwise. For a disrupted entry activity $a = \left(e, e'\right)$ of which the duplicate departure event e' corresponds to a train origin departure, a will be effective in Ω_{dis} if e' is not cancelled:

$$y_a = 1 - c_{e'}, \qquad a = (e, e') \in A_{\text{entry}}^{\text{dis}}, r_{e'} = 1. \tag{10}$$

For a disrupted entry activity $a = \left(e, e'\right)$ of which the duplicate departure event e' does *not* correspond to a train origin departure, we established the following constraints to decide whether a is effective in Ω_{dis}:

$$y_a \le 1 - c_{e'}, \qquad a = (e, e') \in A_{\text{entry}}^{\text{dis}}, r_{e'} = 0, \tag{11}$$

$$y_a \le 1 - s_{a'} + c_{e''} + c_{e'''}, \qquad a = (e, e') \in A_{\text{entry}}^{\text{dis}}, r_{e'} = 0, e''' = \lambda_{e'}, a' = (e'', e''') \in A_{\text{dwell}}^{\text{dis}} \cup A_{\text{pass}}^{\text{dis}}, \tag{12}$$

$$y_a \ge 1 - s_{a'} - c_{e''} - c_{e'''}, \qquad a = (e, e') \in A_{\text{entry}}^{\text{dis}}, r_{e'} = 0, e''' = \lambda_{e'}, a' = (e'', e''') \in A_{\text{dwell}}^{\text{dis}} \cup A_{\text{pass}}^{\text{dis}}. \tag{13}$$

Constraint (11) means that a disrupted entry activity $a = (e, e')$ will not be effective in Ω_{dis} if its corresponding duplicate departure event e' is cancelled. Otherwise, a will be effective only if its corresponding duplicate departure event e' is associated with a *real* stop $a' = (e'', e''') \in A_{\text{dwell}}^{\text{dis}} \cup A_{\text{pass}}^{\text{dis}}$ that has $c_{e''} = 0, c_{e'''} = 0$ and $s_{a'} = 0$ (see Table 4 and Table 5), in which e''' is the departure event corresponding to e': $e''' = \lambda_{e'}$. This is represented by (12) and (13).

5.1.6. Deciding which disrupted boarding activities are effective in Ω_{dis}

Recall that a boarding activity is from a duplicate departure event e to the corresponding departure event e'. For a disrupted boarding activity $a = \left(e, e'\right)$ of which the duplicate departure event e corresponds to a train origin departure, a will be effective in Ω_{dis} if e is not cancelled.:

$$y_a = 1 - c_e, \qquad a = (e, e') \in A_{\text{board}}^{\text{dis}}, r_e = 1. \tag{14}$$

For a disrupted boarding activity $a = \left(e, e'\right)$ of which the duplicate departure event e does *not* correspond to a train origin departure, we decide whether a is effective in Ω_{dis} by

$$y_a \le 1 - c_e, \qquad a = (e, e') \in A_{\text{board}}^{\text{dis}}, r_e = 0, \tag{15}$$

$$y_a \le 1 - s_{a'} + c_e + c_{e'}, \qquad a = (e, e') \in A_{\text{board}}^{\text{dis}}, r_e = 0, a' = (e'', e') \in A_{\text{dwell}}^{\text{dis}} \cup A_{\text{pass}}^{\text{dis}}, \tag{16}$$

$$y_a \ge 1 - s_{a'} - c_{e''} - c_{e'}, \qquad a = (e, e') \in A_{\text{board}}^{\text{dis}}, r_e = 0, a' = (e'', e') \in A_{\text{dwell}}^{\text{dis}} \cup A_{\text{pass}}^{\text{dis}}. \tag{17}$$

Constraint (15) means that a disrupted boarding activity a will not be effective in Ω_{dis} if its corresponding duplicate departure event e is cancelled. Otherwise, a will be effective only if its corresponding departure event e' is associated with a *real* stop $a' = (e'', e') \in A_{\text{dwell}}^{\text{dis}} \cup A_{\text{pass}}^{\text{dis}}$ that has $c_{e''} = 0, c_{e'} = 0$ and $s_{a'} = 0$. This is represented by (16) and (17).

5.1.7. Deciding which disrupted exit activities are effective in Ω_{dis}

Recall that an exit activity is from an arrival event e to an exit event e', which both correspond to the same station. We use a binary parameter f_e with value 1 to indicate that an arrival event e corresponds to a train destination arrival, and 0 otherwise. For a disrupted exit activity $a = \left(e, e'\right)$ of which the arrival event e corresponds to a train destination arrival, a will be effective in Ω_{dis} if e is not cancelled:

$$y_a = 1 - c_e, \qquad a = (e, e') \in A^{\text{dis}}_{\text{exit}}, f_e = 1. \tag{18}$$

For a disrupted exit activity $a = \left(e, e'\right)$ of which the arrival event e' does *not* correspond to a train destination arrival, we established the following constraints to decide whether a is effective in Ω_{dis}:

$$y_a \le 1 - c_e, \qquad a = (e, e') \in A^{\text{dis}}_{\text{exit}}, f_e = 0, \tag{19}$$

$$y_a \le 1 - s_{a'} + c_e + c_{e''}, \qquad a = (e, e') \in A^{\text{dis}}_{\text{exit}}, f_e = 0, a' = (e, e'') \in A^{\text{dis}}_{\text{dwell}} \cup A^{\text{dis}}_{\text{pass}}, \tag{20}$$

$$y_a \ge 1 - s_{a'} - c_e - c_{e''}, \qquad a = (e, e') \in A^{\text{dis}}_{\text{exit}}, f_e = 0, a' = (e, e'') \in A^{\text{dis}}_{\text{dwell}} \cup A^{\text{dis}}_{\text{pass}}, \tag{21}$$

Constraint (19) means that a disrupted exit activity $a = (e, e')$ will not be effective in Ω_{dis} if its corresponding arrival event e is cancelled. Otherwise, a will be effective only if its corresponding arrival event e is associated with a *real* stop $a' = (e, e'') \in A^{\text{dis}}_{\text{dwell}} \cup A^{\text{dis}}_{\text{pass}}$ that has $c_e = 0, c_{e''} = 0$ and $s_{a'} = 0$. This is stated by (20) and (21).

5.1.8. Deciding which disrupted wait activities are effective in Ω_{dis}

Recall that a wait activity is from a duplicate departure event e to the next time-closest duplicate departure event e' that occurs at the same station but corresponds to a different train. We decide whether a disrupted wait activity $a = (e, e')$ is effective in Ω_{dis} by

$$y_a \le 1 - c_e, \qquad a = (e, e') \in A^{\text{dis}}_{\text{wait}}, \tag{22}$$

$$y_a \le 1 - c_{e'}, \qquad a = (e, e') \in A^{\text{dis}}_{\text{wait}}, \tag{23}$$

$$y_a \le 1 - s_{a'} + c_{e''} + c_{e'''}, \qquad a = (e, e') \in A^{\text{dis}}_{\text{wait}}, r_e = 0, e''' = \lambda_e, a' = (e'', e''') \in A^{\text{dis}}_{\text{dwell}} \cup A^{\text{dis}}_{\text{pass}}, \tag{24}$$

$$y_a \le 1 - s_{a'} + c_{e''} + c_{e'''}, \qquad a = (e, e') \in A^{\text{dis}}_{\text{wait}}, r_{e'} = 0, e''' = \lambda_{e'}, a' = (e'', e''') \in A^{\text{dis}}_{\text{dwell}} \cup A^{\text{dis}}_{\text{pass}}, \tag{25}$$

$$y_a + y_{a'} \le 1, \qquad a = (e, e') \in A^{\text{dis}}_{\text{wait}}, a' = (e', e) \in A^{\text{dis}}_{\text{wait}} \tag{26}$$

$$\sum_{\substack{a \in A^{\text{dis}}_{\text{wait}},\\ tail(a) = e}} y_a \le 1, \qquad e \in E^{\text{dis}}_{\text{dde}}, \tag{27}$$

$$\sum_{\substack{a \in A^{\text{dis}}_{\text{wait}},\\ head(a) = e'}} y_a \le 1, \qquad e' \in E^{\text{dis}}_{\text{dde}}, \tag{28}$$

$$x_{e'} - x_{e''} \le M(1 - y_a), \qquad a = (e, e') \in A^{\text{dis}}_{\text{wait}}, a' = (e, e'') \in A^{\text{dis}}_{\text{wait}}, \tag{29}$$

$$x_{e''} - x_{e'} \le M(1 - y_{a'}), \qquad a = (e, e') \in A^{\text{dis}}_{\text{wait}}, a' = (e, e'') \in A^{\text{dis}}_{\text{wait}}, \tag{30}$$

$$x_{e'} - x_e \ge -M(1 - y_a), \qquad a = (e, e') \in A^{\text{dis}}_{\text{wait}}, \tag{31}$$

where *tail*(a) refers to the tail event of an activity: the event which an activity starts from, *head*(a) refers to the head event of an activity: the event which an activity directs to, and M is a sufficiently large number of which the value is set to 2880. Constraints (22) and (23) mean that a disrupted wait activity will not be effective in Ω_{dis} if at least one of the corresponding events is cancelled. Constraint (24) (25) requires a disrupted wait activity $a = (e, e')$ to be *ineffective* if the corresponding duplicate departure event e (e') does not correspond to a train origin departure and is not associated with a real stop. A duplicate departure event could be relevant to multiple disrupted wait activities in a transition network, while at most one of these activities can be effective in an event-activity network Ω_{dis} (26)–(28). Constraints (29)–(31) together ensure that a duplicate departure event e can only be linked to the next time-closest duplicate departure event to construct an effective wait activity in Ω_{dis}.

5.1.9. Deciding which disrupted transfer activities are effective in Ω_{dis}

Recall that a transfer activity is from an arrival event e to the next time-closest duplicate departure event e' that occurs at the same station as e but corresponds to a different train. We decide whether a disrupted transfer activity $a=(e,e')$ is effective in Ω_{dis} by

$$y_a \le 1 - c_e, \qquad a=(e,e') \in A^{\text{dis}}_{\text{trans}}, \tag{32}$$

$$y_a \le 1 - c_{e'}, \qquad a=(e,e') \in A^{\text{dis}}_{\text{trans}}, \tag{33}$$

$$y_a \le 1 - s_{a'} + c_e + c_{e''}, \qquad a=(e,e') \in A^{\text{dis}}_{\text{trans}}, f_e = 0, a'=(e,e'') \in A^{\text{dis}}_{\text{dwell}} \cup A^{\text{dis}}_{\text{pass}}, \tag{34}$$

$$y_a \le 1 - s_{a'} + c_{e''} + c_{e'''}, \qquad a=(e,e') \in A^{\text{dis}}_{\text{trans}}, r_{e'} = 0, e'''=\lambda_{e'}, a'=(e'',e''') \in A^{\text{dis}}_{\text{dwell}} \cup A^{\text{dis}}_{\text{pass}}, \tag{35}$$

$$\sum_{\substack{a\in A^{\text{dis}}_{\text{trans}},\\ tail(a)=e}} y_a \le 1, \qquad e \in E^{\text{trans}}_{\text{ar}} \cup E^{\text{dis}}_{\text{ar}}, \tag{36}$$

$$x_{e'} - x_{e''} \le M(1-y_a), \qquad a=(e,e') \in A^{\text{dis}}_{\text{trans}}, a'=(e,e'') \in A^{\text{dis}}_{\text{trans}}, \tag{37}$$

$$x_{e''} - x_{e'} \le M(1-y_{a'}), \qquad a=(e,e') \in A^{\text{dis}}_{\text{trans}}, a'=(e,e'') \in A^{\text{dis}}_{\text{trans}}, \tag{38}$$

$$x_{e'} - x_e \ge -M(1-y_a) + \ell^{\text{trans}}_{e,e'}, \qquad a=(e,e') \in A^{\text{dis}}_{\text{trans}}, \tag{39}$$

where $\ell^{\text{trans}}_{e,e'}$ refers to the minimum transfer time. Constraints (32) and (33) means that a disrupted transfer activity will not be effective in Ω_{dis} if at least one of the corresponding events is cancelled. Constraint (34) requires a disrupted transfer activity $a=(e,e')$ to be *ineffective* if the corresponding arrival event e does not correspond to a train destination arrival and is not associated with a real stop. Constraint (35) requires a disrupted transfer activity $a=(e,e')$ to be *ineffective* if the corresponding duplicate departure event e' does not correspond to a train origin departure and is not associated with a real stop. Constraint (36) means that for an arrival event $e \in E^{\text{trans}}_{\text{ar}} \cup E^{\text{dis}}_{\text{ar}}$, which has multiple disrupted transfer activities starting from it, at most one of these activities will be effective in an event-activity network Ω_{dis}. Constraints (37)–(39) together ensure that an arrival event e can only be linked to the next time-closest duplicate departure event to construct an effective transfer activity of which the minimum transfer time must be respected.

5.2. Passenger reassignment

There could be multiple passengers who share exactly the same journeys in terms of the planned timetable: the same origin station, the same arrival time at the origin station, the same destination, and the same expected generalized travel time from the origin to the destination. These passengers form a same group $g \in G$, which is assumed to be inseparable in case of a disruption, and will not change the destination. G represents the set of passenger groups, possibly consisting of a single passenger. Recall that a path is a sequence of connected activities. Deciding which path will be chosen by a passenger group is equivalent to deciding which activities will be chosen by this group, while each group g is associated with the same activity choice set A^*. The passenger reassignment module decides which activity $a \in A^*$ will be chosen by a passenger group g and the weight of each activity $a \in A^*$ perceived by g.

5.2.1. Assigning each passenger group to one path only

An activity $a \in A^*$ cannot be chosen by a passenger group if a is *not* effective in Ω_{dis} ($y_a = 0$):

$$u^g_a \le y_a, \qquad a \in A^*, g \in G, \tag{40}$$

where u^g_a is a binary decision with value 1 indicating that activity $a \in A^*$ is chosen by passenger group $g \in G$, and 0 otherwise.

A path that could be chosen by a passenger group g must start from an entry (entry penalty) event corresponding to his/her origin O_g, end in an exit (exit penalty) event corresponding to his/her destination D_g, and include at least one intermediate event to connect them:

$$\sum_{\substack{a\in A^{\text{undis}}_{\text{entry}} \cup A^{\text{dis}}_{\text{entry}} \cup A^{\text{plan}}_{\text{enpenal}},\\ st_{tail(a)}=O_g}} u^g_a = 1, \qquad g \in G, \tag{41}$$

$$\sum_{\substack{a\in A^{\text{undis}}_{\text{entry}} \cup A^{\text{dis}}_{\text{entry}} \cup A^{\text{plan}}_{\text{enpenal}},\\ st_{tail(a)}\neq O_g}} u^g_a = 0, \qquad g \in G. \tag{42}$$

$$\sum_{\substack{a \in A^{\text{undis}}_{\text{exit}} \cup A^{\text{dis}}_{\text{exit}} \cup A^{\text{plan}}_{\text{expenal}}, \\ st_{tail(a)} = D_g}} u^g_a = 1, \qquad g \in G, \tag{43}$$

$$\sum_{\substack{a \in A^{\text{undis}}_{\text{exit}} \cup A^{\text{dis}}_{\text{exit}} \cup A^{\text{plan}}_{\text{expenal}}, \\ st_{tail(a)} \neq D_g}} u^g_a = 0, \qquad g \in G; \tag{44}$$

$$\sum_{a \in In_e} u^g_a = \sum_{a' \in Out_e} u^g_{a'}, \qquad e \in E^* \backslash \left\{E^{\text{plan}}_{\text{entry}}, E^{\text{plan}}_{\text{exit}}\right\}, g \in G, \tag{45}$$

$$M(1 - u^g_a) + x_{e'} \geq t^{\text{ori}}_g, \qquad a = (e, e') \in A^{\text{undis}}_{\text{entry}} \cup A^{\text{dis}}_{\text{entry}}, st_e = O_g, g \in G, \tag{46}$$

where $\{A^{\text{undis}}_{\text{entry}}, A^{\text{dis}}_{\text{entry}}, A^{\text{plan}}_{\text{enpenal}}\}$ contains all entry and entry penality activities in Ω^*, and $\{A^{\text{undis}}_{\text{exit}}, A^{\text{dis}}_{\text{exit}}, A^{\text{plan}}_{\text{expenal}}\}$ contains all exit and exit penalty activities in Ω^*. Recall that an entry (entry penalty activity) is from an entry event to a duplicate departure (penalty) event, while an exit (exit penalty activity) is from an arrival (penalty) event to an exit event. $st_{tail(a)}$ refers to the corresponding station of the tail event of an activity, In_e (Out_e) is the set of activities going in (going out) event e, and t^{ori}_g represents the time of passenger group g arriving at origin station O_g. Constraint (41) means that among the entry and entry penalty activities relevant to the origin of a passenger group, one and only one of them will be chosen by this group. Constraint (42) means that among the entry and entry penalty activities that do *not* correspond to the origin station of a passenger group, *none* of them will be chosen by this group. Constraint (43) means that among the exit and exit penalty activities relevant to the destination of a passenger group, one and only one of them will be chosen by this group. Constraint (44) means that among the exit and exit penalty activities that do *not* correspond to the destination station of a passenger group, *none* of them will be chosen by this group. Constraint (45) is for flow balance at intermediate events (i.e. excluding entry and exit events). It means that if an activity $a = (e', e)$, which goes into an intermediate event e, is chosen by a passenger group g (u^g_a=1), then another activity $a' = (e, e'')$, which goes out from event e should also be chosen by this group ($u^g_{a'}$=1). Constraint (46) means that an entry activity $a = (e, e')$ that corresponds to the origin station of a passenger group could be chosen by this group only if the rescheduled time $x_{e'}$ of the duplicate departure event e' in a is later than the time of this group arriving at the origin station t^{ori}_g.

5.3. Deciding the weight of each activity perceived by a passenger group

Suppose we use w_a to represent the decision on the weight of an activity a. Then, the generalized travel time of a passenger in group g can be described as $\sum_{a \in A^*} w_a \cdot u^g_a$, which is a nonlinear formulation because w_a and u^g_a are both decision variables. To formulate the generalized travel time of a passenger in a linear way, we use w^g_a instead, which is a continuous variable indicating the weight of an activity a perceived by each passenger in group g. The generalized travel time of each passenger in group g is then formulated as $\sum_{a \in A^*} w^g_a$. The value of w^g_a is forced to be 0 if activity a is not chosen by group g. Otherwise, the value of w^g_a is determined according to the time cost of activity a and passenger preference on the type of a. In the following, we introduce the constraints of deciding w^g_a for each kind of activity.

If an activity a is not chosen by group g ($u^g_a = 0$), the weight of this activity will be 0:

$$w^g_a \leq M^* u^g_a, \qquad a = (e, e') \in A^{\text{undis}}_j \cup A^{\text{dis}}_j, \ j \in \{\text{entry, wait, run, dwell, pass, trans}\}, g \in G, \tag{47}$$

$$w^g_a \geq 0, \qquad a = (e, e') \in A^{\text{undis}}_j \cup A^{\text{dis}}_j, \ j \in \{\text{entry, wait, run, dwell, pass, trans}\}, g \in G, \tag{48}$$

where M^* is a sufficiently larger number, of which the value is set to $\beta_{\text{wait}} M$. Here, β_{wait} is the multiplier of waiting time perceived by passengers at stations.

If an entry activity a is chosen by group g ($u^g_a = 1$), the weight of this entry activity perceived by each passenger in group g is determined by

$$w^g_a \leq \beta_{\text{wait}}\left(x_{e'} - t^{\text{ori}}_g\right) + M^*\left(1 - u^g_a\right), \qquad a = (e, e') \in A^{\text{undis}}_{\text{entry}} \cup A^{\text{dis}}_{\text{entry}}, g \in G, \tag{49}$$

$$w^g_a \geq \beta_{\text{wait}}\left(x_{e'} - t^{\text{ori}}_g\right) - M^*\left(1 - u^g_a\right), \qquad a = (e, e') \in A^{\text{undis}}_{\text{entry}} \cup A^{\text{dis}}_{\text{entry}}, g \in G, \tag{50}$$

where w^g_a is forced to be $\beta_{\text{wait}}\left(x_{e'} - t^{\text{ori}}_g\right)$ if $u^g_a = 1$, in which case $x_{e'}$ must be larger than t^{ori}_g due to (46).

If a wait activity a is chosen by group g ($u^g_a = 1$), the weight of this wait activity perceived by each passenger in group g is determined by

$$w^g_a \leq \beta_{\text{wait}}(x_{e'} - x_e) + M^*\left(1 - u^g_a\right), \qquad a = (e, e') \in A^{\text{undis}}_{\text{wait}} \cup A^{\text{dis}}_{\text{wait}}, g \in G, \tag{51}$$

$$w^g_a \geq \beta_{\text{wait}}(x_{e'} - x_e) - M^*\left(1 - u^g_a\right), \qquad a = (e, e') \in A^{\text{undis}}_{\text{wait}} \cup A^{\text{dis}}_{\text{wait}}, g \in G, \tag{52}$$

where w_a^g is forced to be $\beta_{\text{wait}}(x_{e'} - x_e)$ if $u_a^g = 1$, in which case $x_{e'}$ must be larger than x_e (otherwise a would not be effective and then would not be chosen by g).

If a run, dwell or pass-through a is chosen by group g ($u_a^g = 1$), the weight of this activity perceived by each passenger in group g is determined by

$$w_a^g \le \beta_{\text{vehicle}}(x_{e'} - x_e) + M^*\left(1 - u_a^g\right), \qquad a = (e, e') \in A_j^{\text{undis}} \cup A_j^{\text{dis}}, j \in \{\text{run, dwell, pass}\}, g \in G, \tag{53}$$

$$w_a^g \ge \beta_{\text{vehicle}}(x_{e'} - x_e) - M^*\left(1 - u_a^g\right), \qquad a = (e, e') \in A_j^{\text{undis}} \cup A_j^{\text{dis}}, j \in \{\text{run, dwell, pass}\}, g \in G, \tag{54}$$

where β_{vehicle} is the multiplier of in-vehicle time perceived by passengers. Here, w_a^g is forced to be $\beta_{\text{vehicle}}(x_{e'} - x_e)$ if u_a^g=1, in which case $x_{e'}$ must be larger than x_e (otherwise a would not be effective and then would not be chosen by g).

If a transfer activity a is chosen by group g ($u_a^g = 1$), the weight of this transfer activity perceived by each passenger in group g is determined by

$$w_a^g \le \beta_{\text{trans}} + \beta_{\text{wait}}(x_{e'} - x_e) + M^*\left(1 - u_a^g\right), \qquad a = (e, e') \in A_{\text{trans}}^{\text{undis}} \cup A_{\text{trans}}^{\text{dis}}, g \in G, \tag{55}$$

$$w_a^g \ge \beta_{\text{trans}} + \beta_{\text{wait}}(x_{e'} - x_e) - M^*\left(1 - u_a^g\right), \qquad a = (e, e') \in A_{\text{trans}}^{\text{undis}} \cup A_{\text{trans}}^{\text{dis}}, g \in G, \tag{56}$$

where β_{trans} is the fixed time penalty of one transfer. Here, w_a^g is forced to be $\beta_{\text{trans}} + \beta_{\text{wait}}(x_{e'} - x_e)$ if u_a^g=1, in which case $x_{e'}$ must be larger than x_e (otherwise a would not be effective and then would not be chosen by g).

The weight of an entry penalty activity is determined by

$$w_a^g = \mu \cdot T_g^{\text{plan}} \cdot u_a^g, \qquad a = (e, e') \in A_{\text{enpenal}}^{\text{plan}}, g \in G, \mu \ge 1, \tag{57}$$

where T_{plan}^g refers to the expected generalized travel time of passenger group g in terms of the planned timetable, and $\mu \cdot T_{\text{plan}}^g$ refers to the maximum generalized travel time which passenger group g would accept during a disruption. Both T_{plan}^g and μ are given parameters.

The weight of a boarding activity, an exit activity, or an exit penalty activity is set to 0:

$$w_a^g = 0, \qquad a = (e, e') \in \left\{A_{\text{expenal}}^{\text{plan}},\ A_{\text{board}}^{\text{undis}},\ A_{\text{board}}^{\text{dis}},\ A_{\text{exit}}^{\text{undis}},\ A_{\text{exit}}^{\text{dis}}\right\}, g \in G. \tag{58}$$

5.4. Objective

The objective is to minimize the generalized travel times of all passengers, which is

$$z_p = \sum_{g \in G} \sum_{a \in A^*} n_g w_a^g, \tag{59}$$

where n_g represents the number of passengers in group g.

To summarize, the proposed passenger-oriented timetable rescheduling model (POTR) is given by constraints (1)–(58) presented in this paper, as well as the constraints presented in Zhu and Goverde (2019c), with objective (59).

6. Reducing the computational complexity of the passenger-oriented timetable rescheduling model

When dealing with a large railway network and/or considering numerous passengers, the proposed passenger-oriented timetable rescheduling model (POTR) may not be able to find a high-quality solution in an acceptable time, because a binary variable u_a^g is created for each activity $a \in A^*$ associated with each passenger group $g \in G$, of which the total number is $|A^*| \times |G|$. To reduce the computational complexity, we propose 1) a pre-processing method to shrink the activity choice set for each passenger group in a reasonable way, and 2) an iterative solution method to solve the model with limited passenger groups considered in each iteration, which also restricts the solution space to avoid excessive operational deviations that are not preferred by the railway operators. We introduce both methods in this section.

6.1. Shrinking the activity choice set of a passenger group

The passenger-oriented timetable rescheduling model proposed in Section 5 considers A^* as the activity choice set for each passenger group $g \in G$, while A^* contains some activities that will never be chosen by g. Thus, we introduce a method of constructing an improved activity choice set A_g^* for passenger group $g \in G$ by excluding the activities that will never be chosen by g. In other words, $A_g^* \subset A^*$, and $\bigcup_g A_g^* \subseteq A^*$. Recall that $A^* = A_{\text{run}}^{\text{plan}} \cup A_{\text{dwell}}^{\text{plan}} \cup A_{\text{pass}}^{\text{plan}} \cup A_{\text{wait}}^* \cup A_{\text{trans}}^* \cup A_{\text{board}}^* \cup A_{\text{entry}}^* \cup A_{\text{exit}}^* \cup A_{\text{enpenal}}^{\text{plan}} \cup A_{\text{expenal}}^{\text{plan}}$. Table 6 shows activity sets relevant to passenger group g.

We first define the events that could be reachable by passenger group g, and then define the activity set A_g^*, which contains all activities that could be chosen by g according to the reachable events. We decide whether event e is reachable by group g in terms of multiple factors. These factors are the time of passenger group g arriving at the origin, t_g^{ori}; the

Table 6
Activity sets relevant to passenger group g.

Notation	Description
A^*_g	The activity choice set associated with passenger group g: $A^*_g \subset A^*$
$A^{\text{plan}}_{i,g}$	Set of i activities associated with passenger group g: $A^{\text{plan}}_{i,g} \subset A^{\text{plan}}_i$, $i \in \{\text{run, dwell, pass}\}$
$A^*_{k,g}$	Set of k activities associated with passenger group g: $A^*_{k,g} \subset A^*_k$, $k \in \{\text{wait, trans, board, entry.exit}\}$

maximum acceptable generalized travel time of passenger group g, μT^{plan}_g; the minimal travel time from the origin of group g to the corresponding station of event e, $\ell^{\min}_{O_g, st_e}$; the minimal travel time from the corresponding station of event e to the destination of group g, $\ell^{\min}_{st_e, D_g}$; the maximum delay allowed per train event, D; and the original scheduled time of event e, o_e.

Event e is defined reachable by passenger group g if the following two conditions are *both* satisfied:

1. $o_e + D \geq t^{\text{ori}}_g + \ell^{\min}_{O_g, st_e}$,
2. $o_e \leq t^{\text{ori}}_g + \mu T^{\text{plan}}_g - \ell^{\min}_{st_e, D_g}$.

In condition 1, $o_e + D$ represents the maximum rescheduled time of event e, which should be later than the *earliest* time point when g could reach the station st_e from the origin. Otherwise, e will not be reachable by g. Condition 2 means that the minimum rescheduled time of event e (equivalent to the value of o_e) should be earlier than the *latest* time point that g can be at station st_e so that he/she reaches the destination below the acceptable delay threshold. Otherwise, e will not be reachable by g.

Based on the defined reachable events, we add

- an activity $a = (e, e') \in A^{\text{plan}}_i$ to $A^{\text{plan}}_{i,g}$ for any $i \in \{\text{run, dwell, pass}\}$, if *both* events e and e' could be reachable by passenger group g,
- an activity $a = (e, e') \in A^*_i$ to $A^*_{j,g}$ for any $j \in \{\text{wait, trans, board}\}$, if *both* events e and e' could be reachable by passenger group g,
- an activity $a = (e, e') \in A^*_{\text{entry}}$ to $A^*_{\text{entry},g}$ if duplicate departure event e' could be reachable by passenger group g and $st_{e'} = O_g$,
- an activity $a = (e, e') \in A^*_{\text{exit}}$ to $A^*_{\text{exit},g}$ if arrival event e could be reachable by passenger group g and $st_{e'} = D_g$, and
- each activity $a \in A^{\text{plan}}_{\text{enpenal}} \cup A^{\text{plan}}_{\text{expenal}}$ to A^*_g.

Thus, $A^*_g = \{A^{\text{plan}}_{i,g}\}_{i \in I} \cup \{A^*_{k,g}\}_{k \in K} \cup A^{\text{plan}}_{\text{enpenal}} \cup A^{\text{plan}}_{\text{expenal}}$, in which $I = \{\text{run, dwell, pass}\}, K = \{\text{wait, trans, board, entry, exit}\}$. The constructed A^*_g reduces the number of binary variables u^g_a and continuous variables w^g_a, as well as the corresponding constraints in the passenger-oriented timetable rescheduling model.

Using Fig. 2 as an example we show which activities should be excluded for a specific passenger group. Suppose passenger group g plans to travel from station A (O_g) to station C (D_g), and arrives at the origin station A after the planned departure of train tr_1 but before train tr_2. In this case we assume that the events that can *not* be reachable by g include (dde, tr_1, A), (de, tr_1, A), and (ar, tr_1, B), which depend on the values of specific factors (e.g., t^{ori}_g, D, μT^{plan}_g, etc.) as explained before. The activities, which consist of at least one of the non-reachable events will be excluded from the activity choice set A^*_g of passenger group g. These include the entry activity directing to event (dde, tr_1, A), the waiting activities between events (dde, tr_1, A) and (dde, tr_2, A), the boarding activity from (dde, tr_1, A) to (de, tr_1, A), the running activity from (de, tr_1, A) to (ar, tr_1, B), and the transfer activity from (ar, tr_1, B) to (dde, tr_2, B). Besides, all entry activities to stations B and C, and all exit activities from stations A and B are *also* excluded from A^*_g.

6.2. Adapted fix-and-Optimize (AFaO) algorithm

The Fix-and-Optimize (FaO) algorithm iteratively solves a model over all real-valued variables with the majority of the binary variables fixed (Sahling et al., 2009; Lang and Shen, 2011; Franz et al., 2019), which is used and adapted in this paper as the *Adapted* Fix-and-Optimize (AFaO) (Algorithm 1). The AFaO algorithm is different from the FaO in the sense that AFaO includes less variables in an earlier iteration, while FaO includes all variables in each iteration. Our AFaO solves the passenger-oriented timetable rescheduling model iteratively by considering limited passenger groups in each iteration, where the timetable rescheduling problem is solved for all train services although restricted passenger groups are considered. Each iteration determines the activities for the next set of additional considered groups while keeping the activities of the previous considered groups as fixed. More details about the proposed AFaO is introduced below.

The number of new passenger groups n_{new} considered in an iteration determines the number of required iterations $I = \lceil \frac{|G|}{n_{\text{new}}} \rceil$, where G represents the set of all passenger groups. G is sorted according to a specific rule that decides the order of passengers to be considered. The sorting rule affects the solution quality. In Section 7.3.4 we carried out extensive numerical experiments to investigate the impacts of different sorting rules on the solution quality. The impacts of n_{new} on

Algorithm 1: The adapted fix-and-optimize algorithm.

Input: OOTR, POTR, G, E^*, $\left\{A_g^*\right\}_{g\in G}$, n_{new}, t^{stop}, T^{stop}, Δ

1 Solve the OOTR model to get z_o^*;
2 Add constraint $\sum_{e\in E_{\text{ar}}^{\text{plan}}} w_c c_e + d_e \le z_o^* + \Delta$ to the POTR model;
3 $I = \left\lceil \frac{|G|}{n_{\text{new}}} \right\rceil$;
4 $G' = \emptyset$;
5 $\text{POTR}^1 = \text{POTR}$;
6 $i = 1$;
7 **while** *$i \le I$ and T^{stop} is not reached* **do**
8   **if** $i < I$ **then**
9     $G_{\text{new}} = \{g_{j+1}, \cdots, g_{j+n_{\text{new}}}\} \subset G$, $j = (i-1)n_{\text{new}}$;
10     $G' = G' \cup G_{\text{new}}$;
11     $\left\{\tilde{x}_e, \tilde{u}_a^g, \tilde{z}_p\right\} \leftarrow$ solve POTR^i for all $g \in G'$ within t^{stop} or until a feasible solution is found, using the solution of OOTR model as an initial guess if $i = 1$ or the solution of POTR^{i-1} as an initial guess if $i \ge 2$;
12   Construct POTR^{i+1} by adding constraints: $u_a^g = \tilde{u}_a^g$, $a \in A_g^*$, $g \in G_{\text{new}}$, into POTR^i ;
13     $i = i + 1$;
14   **else**
15     $G_{\text{new}} = \left\{g_{(i-1)n_{\text{new}}+1}, \cdots, g_{|G|}\right\} \subset G$;
16     $G' = G' \cup G_{\text{new}}$;
17     $\left\{\tilde{x}_e, \tilde{u}_a^g, \tilde{z}_p\right\} \leftarrow$ solve POTR^i for all $g \in G$ within t^{stop} or until a feasible solution is found, using the solution of POTR^{i-1} as an initial guess;
18 **if** $G' \neq G$ **then**
19   Construct POTR' by adding constraints: $x_e = \tilde{x}_e$, $e \in E_{\text{ar}}^{\text{plan}} \cup E_{\text{de}}^{\text{plan}} \cup E_{\text{dde}}^*$, into POTR ;
20   $\left\{\tilde{x}_e, \tilde{u}_a^g, \tilde{z}_p\right\} \leftarrow$ solve POTR' for all $g \in G$;
21 **Return** $\left\{\tilde{x}_e, \tilde{u}_a^g, \tilde{z}_p\right\}$ finally obtained;

the solution quality and the computation time of the iterative solution method are also investigated in Section 7.3. The iterative solution method terminates until either all passenger groups in G are considered or the total running time limit T^{stop} is reached, while in the latter case one more process will be needed to evaluate the responses of all passengers towards the rescheduled timetable finally obtained. At each iteration a computation time limit t^{stop} is set to avoid excessive searching for the optimal solution, while a longer computation than t^{stop} will be allowed to find a feasible solution in case no feasible solution can be obtained within t^{stop}.

Algorithm 1 needs the following inputs: the operator-oriented timetable rescheduling model (OOTR), the passenger-oriented timetable rescheduling model (POTR), the set of passenger groups G, the set of events E^*, the activity choice set A_g^* of each passenger group $g \in G$, the number of new passengers n_{new} considered in each iteration, the computation time limit t^{stop} of each iteration, the total computation time limit T^{stop}, and the maximum allowed deviation Δ from the optimal operator-oriented objective value in terms of train cancellations and delays. The OOTR model is the timetable rescheduling module from Zhu and Goverde (2019c), which in this paper adopts the objective of minimizing train cancellations and delays: $z_o = \sum_{e\in E_{\text{ar}}^{\text{plan}}} w_c c_e + d_e$, where c_e is a binary cancellation decision, d_e is a continuous decision representing the delay of event e, and w_c is the penalty of cancelling a train service between two neighbouring stations. Solving the OOTR model to optimality gets the optimal operator-oriented objective z_o^*. The POTR model consists of the timetable rescheduling module, the dynamic event-activity network formulation module, and the passenger reassignment module, which aims to minimize the generalized travel times of passengers: $z_p = \sum_{g\in G} \sum_{a\in A_g^*} n_g w_a^g$.

In Algorithm 1, the OOTR model is solved first to obtain the optimal operator-oriented rescheduled timetable that has the operator-oriented objective value of z_o^* (line 1). To find a passenger-friendly rescheduled timetable that can also be preferred by railway operators, we add a constraint to the POTR model to require that the passenger-oriented rescheduled timetable obtained by the POTR model will not deviate from the optimal operator-oriented rescheduled timetable by Δ in terms of train cancellations and delays (line 2). Line 3 initializes the number of iterations needed to solve the POTR model, and line 4 initializes the set of passenger groups considered at each iteration as an empty set. In line 5, we define the passenger-oriented timetable rescheduling model to be solved in the 1st iteration as POTR^1. The iteration is initialized in line 6. If the required iterations are not completely performed and the total computation time until the current iteration is shorter than T^{stop}, then the while-loop starting from line 7 continues.

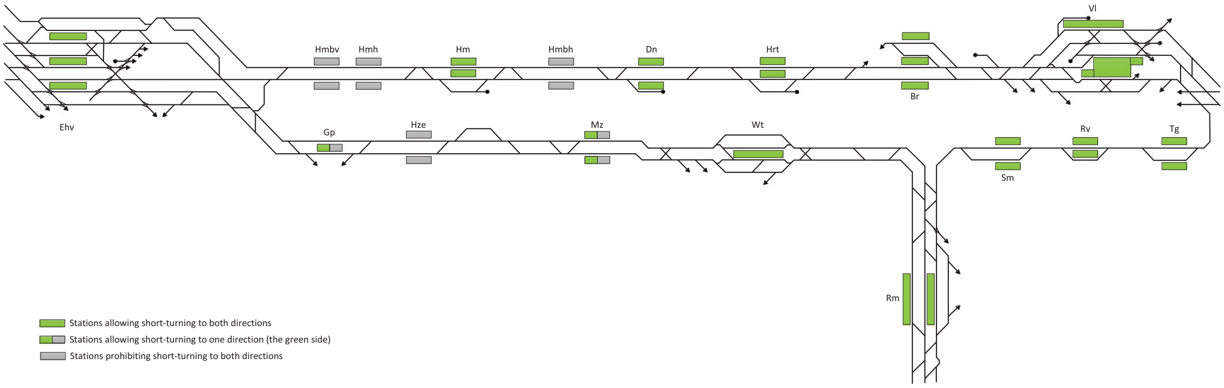


Fig. 3. The schematic track layout in the considered network.

If the current iteration is not the final iteration (line 8), we select n_{new} passenger groups from G as the passenger groups that are *newly* considered in the current iteration (line 9), which are then added to G' (line 10). Considering all passenger groups in G', the current POTR^i model is solved within the required time limit t^{stop}. If no feasible solution has been found within t^{stop}, the search will continue until a feasible solution is found (line 11). To speed up the computation time, we give an initial guess to warm-start the solver. Note that GUROBI can benefit from an initial guess on where to start the search, so called warm-starting. When the current iteration is the first iteration, the initial guess is chosen as the solution obtained earlier from the OOTR model. When the current iteration is the second or a later iteration, the initial guess is chosen as the solution from the POTR^{i-1} model in the previous iteration. The outputs of the current model POTR^i include the rescheduled time $\tilde{x}_e$ of event $e \in E_{\text{ar}}^{\text{plan}} \cup E_{\text{de}}^{\text{plan}} \cup E_{\text{dde}}^{*}$, the choice $\tilde{u}_a^g$ of passenger group $g \in G'$ on activity $a \in A_g^*$, and the passenger-oriented objective value $\tilde{z}_p$ that represents the generalized travel times over all passenger groups in G'. In line 12, we construct the passenger-oriented timetable rescheduling model POTR^{i+1} to be solved in the next iteration by adding constraints to the current POTR^i. These constraints require the activity choices of the passenger groups that are newly considered in the current iteration to be fixed to the assigned paths in all following iterations. In line 13, we proceed to the next iteration.

If the current iteration is the final iteration (line 14), we add the passenger groups that have not be considered yet (line 15) to the G' (line 16), which now includes all passenger groups of G: $G' = G$. Considering all passenger groups in G, the current POTR^i model is solved within the required time limit t^{stop}. If no feasible solution has been found within t^{stop}, the search will continue until the first feasible solution is found by giving the solution from the previous iteration as an initial guess to warm-start the solver (line 17). In this case, the algorithm terminates by returning the results from POTR^i (lines 21).

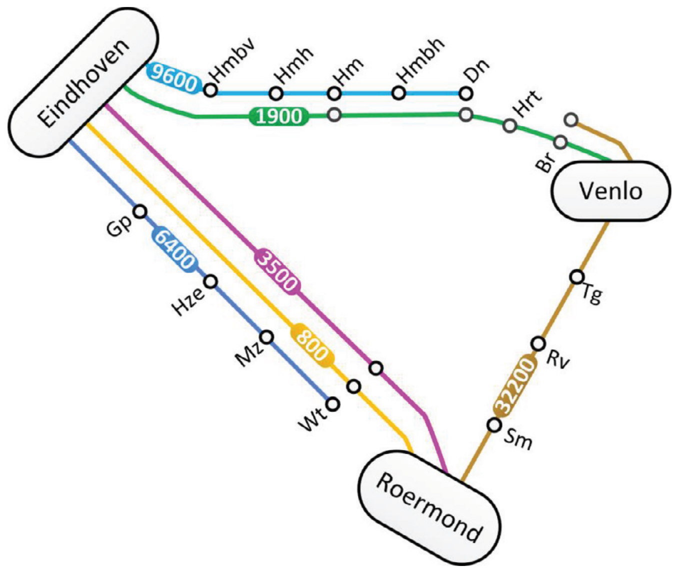
The while-loop could end before all required iterations are performed due to the total computation limit of T^{stop}. Under this circumstance, the passenger groups in G have not been completely considered (line 18), which means that the $\tilde{z}_p$ finally obtained in the while-loop does not represent the generalized travel times of all passenger groups in G. Therefore, we construct POTR$'$ by adding constraints to the original POTR model, which require the rescheduled timetable finally obtained in the while-loop to be fixed (line 19). In that sense, solving POTR$'$ is *not* to compute a new rescheduled timetable but to evaluate the generalized travel times of all passenger groups in G under a given rescheduled timetable that is finally obtained in the while-loop (line 20). Hence, the computation time of solving POTR$'$ is not counted in T^{stop}.

7. Case study

The case study aims to investigate the performance of the passenger-oriented timetable rescheduling model on shortening generalized travel times during railway disruptions, and to analyse the computational efficiency of the proposed AFaO algorithm to the passenger-oriented timetable rescheduling model. Section 7.1 describes the case study, while Section 7.2 and Section 7.3 report the performance of the passenger-oriented model and the computational efficiency of the proposed algorithm, respectively.

7.1. Setup

The case study is performed to a part of the Dutch railways, of which the schematic track layout is shown in Fig. 3. The considered network is totally around 128 km long, which has both single-track (23.5 km) and double-track (104.5 km) railway lines with in total 17 stations. The stations that allow short-turning to both directions are colored in full green, the stations that prohibit short-turning to both directions are colored in full grey, and the stations that allow (prohibit) short-turning to one direction are colored in half green (grey). Six train lines operate half-hourly in each direction in the considered network, of which the scheduled stopping patterns are indicated in Fig. 4, as well as the terminal stations of



Train line	Terminal
IC800	—
IC1900	Venlo (Vl)
IC3500	—
SPR6400	Wt and Eindhoven (Ehv)
SPR9600	Dn
SPR32200	Roermond (Rm)

Fig. 4. The train lines operating in the considered network.

Table 7
Parameter settings.

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
$\ell^{\text{trans}}_{e,e'}$	5 min	β_{trans}	10 min	D	25 min	t^{stop}	30 s
$\ell^{\max}_{\text{trans}}$	30 min	β_{vehicle}	1	R	25 min	T^{stop}	300 s
$\ell^{\max}_{\text{wait}}$	30 min	β_{wait}	2.5	μ	2		

these train lines in the considered network. The rolling stock circulations at the short-turning and terminal stations of trains are both dealt with. We distinguish between intercity (IC) and local (called sprinter (SPR) in Dutch) train lines. All experiments were carried out in MATLAB on a desktop with Intel Xeon CPU E5-1620 v3 at 3.50 GHz and 16 GB RAM. The solver GUROBI release 7.0.1 was used either to solve the passenger-oriented timetable rescheduling model directly or called by the iterative solution method to solve the model iteratively.

The parameters used to construct an event-activity/transition network $\ell^{\text{trans}}_{e,e'}$, $\ell^{\max}_{\text{trans}}$ and $\ell^{\max}_{\text{wait}}$ are set to 5 min, 30 min and 30 min, respectively. Recall that $\ell^{\text{trans}}_{e,e'}$ represents the minimum transfer time, and $\ell^{\max}_{\text{trans}}$ ($\ell^{\max}_{\text{wait}}$) represents the maximum transfer (waiting) time which a passenger is willing to spend at a station. The maximum delay allowed to a train departure/arrival D is set to 25 min. The disruption timetable is required to recover to the planned timetable no later than 25 min after the disruption ends: $R = 25$. Passengers are assumed to leave the railways if they cannot find paths with less than two times of their expected generalized travel times within the railways: $\mu = 2$. The coefficient of waiting time at an origin/transfer station β_{wait} is set to 2.5 and the coefficient of in-vehicle time β_{vehicle} is set to 1 (Wardman, 2004). The penalty of one transfer β_{trans} is set to 10 min (de Keizer et al., 2012). For the iterative solution method to the passenger-oriented timetable rescheduling model, we set the total computation time limit T^{stop} to 300 s, and the computation time limit of each iteration t^{stop} to 30 s. Table 7 lists the parameter values.

We consider four cases with increasing disruption length. We consider the passengers whose arrival times at the origin stations are during the period of $[t_{\text{start}}, t_{\text{end}} + R]$. Note that although passengers who started travel before t_{start} and are still travelling when the disruption starts are not considered in the case study, they could be handled still. This requires a pre-processing step to determine the first arrival stations of these passengers during the disruption, and then given as input to the proposed passenger-oriented model, of which the formulation does not need to change. We form the passengers who share the same expected journey in terms of the planned timetable into the same group $g \in G$. The number of passenger groups $|G|$ varies with the disruption starting/ending time and the required recovery time length. Table 8 indicates the total numbers of passenger groups and the total numbers of passengers considering different disruption durations but the same required recovery time length 25 min. To show the size of the problem considered in each cases, the numbers of continuous variables, binary variables, and constraints are also given, which are associated with disrupted section Mz-Hze considering $\Delta = 10$ as an example. Note that he numbers of passengers in different groups can be different. In each case of Table 8, the largest group contains 126 passengers, while the smallest group contains 1 passenger only. Fig. 5 shows the numbers of passenger groups considering different group sizes. Recall that n_g refers to the number of passengers in a group g. In each case, most groups contain less than 10 passengers, and few groups contain 30 passengers or more.

Table 8
Disruption and passenger demand cases.

Case	Disruption start	Disruption end	Travel starting period	Total number of passenger groups $\|G\|$	Total number of passengers $\sum_{g\in G} n_g$	Continuous* variables	Binary* variables	Constraints*
I	8:00	8:30	[8:00,8:30+00:25]	334	2,557	189,017	215,783	1,240,814
II	8:00	9:00	[8:00,9:00+00:25]	477	3,320	314,752	342,325	2,011,421
III	8:00	9:30	[8:00,9:30+00:25]	618	3,897	437,067	465,452	2,761,470
IV	8:00	10:00	[8:00,10:00+00:25]	728	4,357	538,420	567,622	3,384,069

*The variables and constraints associated with disrupted section Mz-Hze considering Δ=10

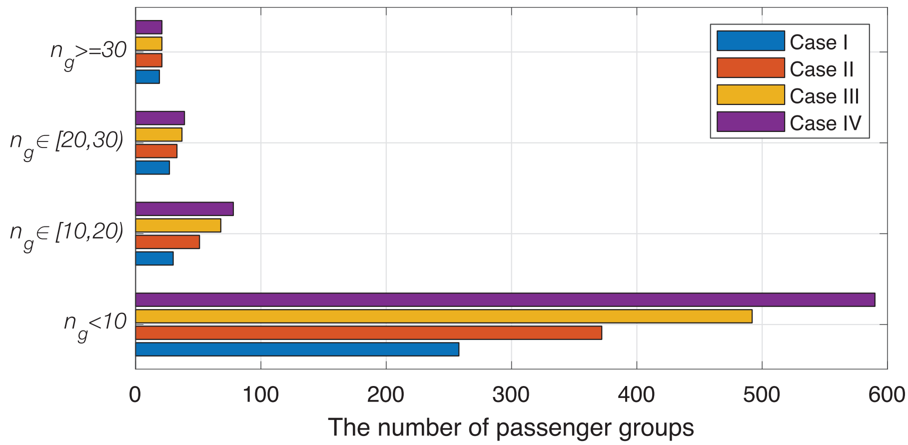


Fig. 5. Passenger group sizes.

Table 9
Notation relevant to a passenger-oriented (operator-oriented) solution.

Notation	Description
z_o	The objective of the OOTR model: minimizing train cancellations and arrival delays $z_o = \sum_{e\in E_{\mathrm{ar}}^{\mathrm{plan}}} w_c c_e + d_e$
w_c	The penalty of cancelling a train service between neighbouring stations: $w_c = 100$
z_p	The objective of the POTR model: minimizing the generalized travel times over all passengers $z_p = \sum_{g\in G}\sum_{a\in A_g^*} n_g w_a^g$
z_o^*	The objective value of the *optimal* rescheduled timetable obtained by the OOTR model
$\tilde{z}_o$	The resulting train cancellations and arrival delays of a rescheduled timetable obtained by the POTR model $\tilde{z}_o = \sum_{e\in E_{\mathrm{ar}}^{\mathrm{plan}}} w_c c_e + d_e,$
$\tilde{z}_p$	The resulting generalized travel times over all passengers of a rescheduled timetable obtained by the OOTR model $\tilde{z}_p = \sum_{g\in G}\sum_{a\in A_g^*} n_g w_a^g$
Δ	The maximum allowed deviation of $\tilde{z}_o$ from z_o^*

We consider section Mz-Hze (between Eindhoven and Roermond) to be completely blocked during the considered four disruption periods (see Table 8), respectively. The operator-oriented timetable rescheduling model (OOTR) is adopted from Zhu and Goverde (2019c), which uses the objective z_o of minimizing train cancellations and arrival delays in this paper. The OOTR is used to solve each disruption case to obtain optimal z_o^*. The passenger-oriented timetable rescheduling model (POTR), which uses the objective z_p of minimizing generalized travel times, is used to solve each disruption case by requiring that the resulting train cancellations and arrival delays cannot exceed $z_o^* + \Delta$, where $\Delta \geq 0$. The resulting generalized travel times of the rescheduled timetables obtained by the OOTR model are evaluated and denoted by $\tilde{z}_p$. The resulting train cancellations and arrival delays of the rescheduled timetables obtained by the POTR model are evaluated and denoted by $\tilde{z}_o$. Table 9 gives the notation relevant to a passenger-oriented (operator-oriented) solution. Note that the penalty w_c of cancelling one service is set to 100.

7.2. The performance of the passenger-oriented timetable rescheduling model

Table 10 shows the optimal solutions obtained from both the operator-oriented and the passenger-oriented timetable rescheduling models by using Gurobi directly (i.e. not using the AFaO algorithm). Due to different objectives, the optimality gap of a solution obtained by the operator-oriented model is different from the optimality gap of a solution obtained by the passenger-oriented model, and thereby we use 'O-gap (z_o)' and 'O-gap (z_p)' to distinguish them. In each case, the passenger-

Table 10
General results by using a solver directly: disrupted section Mz-Hze.

Case	Operator-oriented (solver)				Passenger-oriented (solver): Δ=10					
	z_o [min]	$\tilde{z}_p$ [min]	Time [sec]	O-gap (z_o) [%]	$\tilde{z}_o$ [min]	z_p [min]	Time [sec]	O-gap (z_p) [%]	$z_o - \tilde{z}_o$ [min]	$z_p - \tilde{z}_p$ [min]
I	848	116,211	5	0.00	857	110,757	28	0.00	9	-5,454
II	2,742	161,057	7	0.00	2,752	154,568	150	0.00	10	-6,489
III	4,639	195,944	10	0.00	4,649	189,322	370	0.00	10	-6,622
IV	6,536	221,773	12	0.00	6,546	216,481	250	0.00	10	-5,292

Table 11
Train-related results by using a solver directly: disrupted section Mz-Hze.

Case	Operator-oriented (solver)				Passenger-oriented (solver): Δ=10			
	# Cancelled services	Train arrival delays [min]	# Extra stops	# Skipped stops	# Cancelled services	Train arrival delays [min]	# Extra stops	# Skipped stops
I	6	248	3	2	6	257	6	4
II	24	342	3	2	24	352	6	6
III	42	439	3	2	42	449	6	7
IV	60	536	4	2	60	546	5	8

Table 12
Passenger-related results by using a solver directly: disrupted section Mz-Hze.

Case	Operator-oriented (solver)					Passenger-oriented (solver): Δ=10				
	# Dropped groups	# Dropped passengers	Total in-vehicle time [min]	Total wait wait [min]	Total # transfer	# Dropped groups	# Dropped passengers	Total in-vehicle time [min]	Total wait time [min]	Total # transfer
I	21	228	45,205	21,975	85	18↓	200 ↓	45,824 ↑	20,004 ↓	83↓
II	54	593	49,527	27,654	92	45↓	514 ↓	51,714 ↑	25,683 ↓	91↓
III	88	820	53,612	31,544	165	81↓	800 ↓	54,126 ↑	29,145 ↓	164↓
IV	126	1,028	56,710	33,754	176	123↓	1,025 ↓	56,685 ↓	31,723 ↓	176

oriented solution reduced generalized travel times ($z_p - \tilde{z}_p$) by over 5000 min with at most 10 min additional train delay ($\tilde{z}_o - z_o$) than the optimal operator-oriented solution when setting Δ to 10 in the passenger-oriented model.

Table 11 gives train-related results in more detail. In each case, the numbers of cancelled services were the same in the operator-oriented and the passenger-oriented solutions. This is because the deviation of a passenger-oriented solution from the optimal operator-oriented solution cannot exceed 10 ($\Delta = 10$) while the penalty of cancelling one train service is set to 100 (w_c=100). The resulting total train arrival delays were different in the operator-oriented and the passenger-oriented solutions, as well as the numbers of extra stops and skipped stops. The number of both extra stops and skipped stops in a passenger-oriented solution was more than in the operator-oriented solution of the same case, because larger operation deviation was allowed in the passenger-oriented model which thereby made more changes on train stopping patterns to reflect on passenger needs. We want to emphasize that in the passenger-oriented model the decisions of adding or skipping stops were made with the aim of reducing generalized travel times, whereas in the operator-oriented model these decisions were made with the aim of reducing train cancellations and arrival delays. For example in the operator-oriented model an extra stop will be added to a train at the station where this train was originally planned to pass through but now has to dwell at this station for waiting on platform capacity to be released in a downstream station where this train will be short-turned.

Table 12 gives passenger-related results in more detail, where the symbol ↓ (↑) is used to denote the decrease (increase) in a passenger-oriented solution compared to the operator-oriented solution of the same case. We can see that compared to the operator-oriented solutions, the passenger-oriented solutions resulted in less passenger groups leaving the railways, and the total number of passengers in these groups was also smaller. The passenger-oriented solutions also helped to shorten passenger waiting times at stations in all cases and reduce the number of transfers in most cases. In cases I-III, the passenger-oriented solutions resulted in longer passenger in-vehicle times, because in the passenger-oriented objective waiting times at stations were penalized 2.5 times of in-vehicle times considering passenger preferences (β_{wait}=2.5 and β_{vehicle}=1). Under this circumstance, the passenger-oriented model tends to delay the departures of specific trains at specific stations, which is beneficial to passengers who could now catch the train. The waiting times of these passengers were reduced by earlier boarding because of the delayed train departures, whereas other passengers who were on-board the delayed trains experienced longer in-vehicle times.

To investigate the impact of maximum allowed operation deviation Δ on the solutions obtained by the passenger-oriented model, we performed 10 more experiments on case IV using different values of Δ. The results are shown in Fig. 6. Each green triangle indicates the performance of a solution obtained by the passenger-oriented model using a specific Δ,

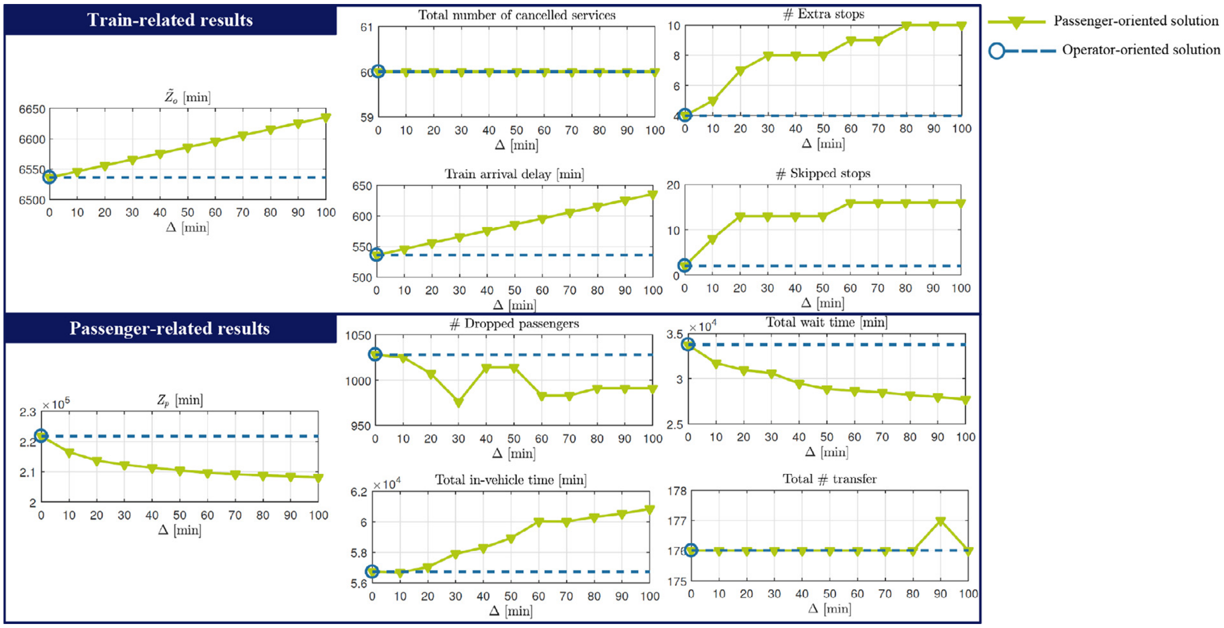


Fig. 6. The optimal passenger-oriented solutions for case IV under different settings of Δ: disrupted section Mz-Hze.

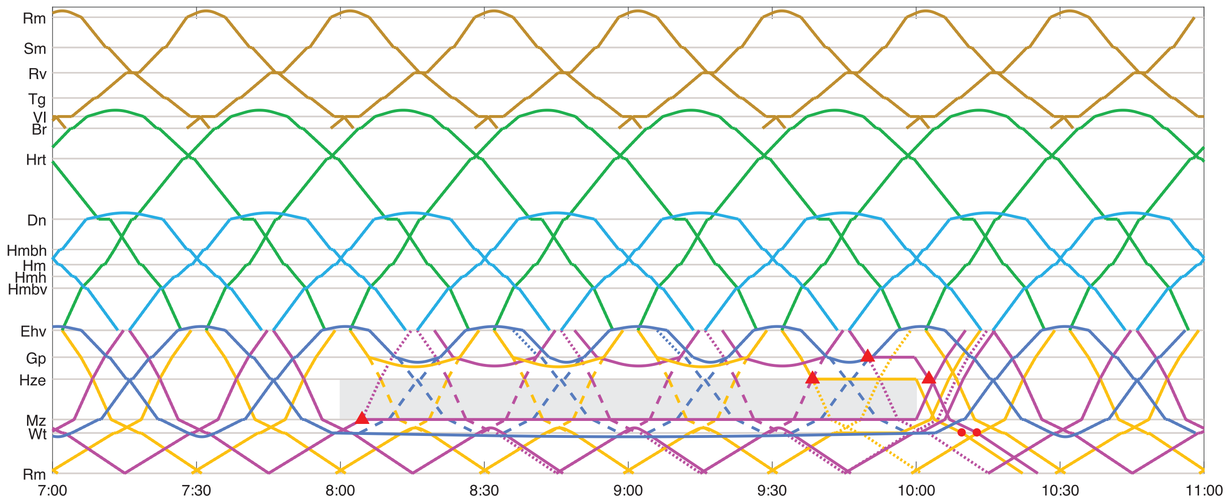


Fig. 7. The optimal operator-oriented rescheduling solution for case IV: disrupted section Mz-Hze.

and each blue circle indicates the performance of the optimal solution obtained by the operator-oriented model. On the one hand, with the increase of Δ the passenger-oriented model resulted in larger weighted train cancellations and train arrival delays ($\tilde{z}_o$). The number of cancelled services always remained the same while train arrival delays increased gradually with the growth of Δ. More extra stops and skipped stops were created under larger Δ. On the other hand, with larger operation deviation allowed the passenger-oriented model resulted in shorter generalized travel times (z_p). The generalized travel time of a passenger is the sum of the weighted waiting time, in-vehicle time and the number of transfers. It can be seen that larger Δ led to shorter waiting times but longer in-vehicle times, whereas the number of transfers almost remained the same. This is because waiting time is perceived 2.5 times of the same length of in-vehicle time by passengers so that the model tends to reduce the waiting times of some passengers at the expense of longer in-vehicle times of other passengers. Under whichever Δ, the number of passengers who chose to leave the railways was always smaller in a passenger-oriented solution compared to the operator-oriented solution.

We take case IV as an example to show the operator-oriented solution in Fig. 7, and the passenger-oriented solutions of $\Delta = 10$ and $\Delta = 30$ in Fig. 8 and Fig. 9, respectively. The grey rectangle indicates the time-distance disruption window, the dashed (dotted) lines represent the original scheduled services that were cancelled (delayed), the solid lines represent the services scheduled in the rescheduling solution, and the red triangles (circles) represent the extra (skipped) stops. The

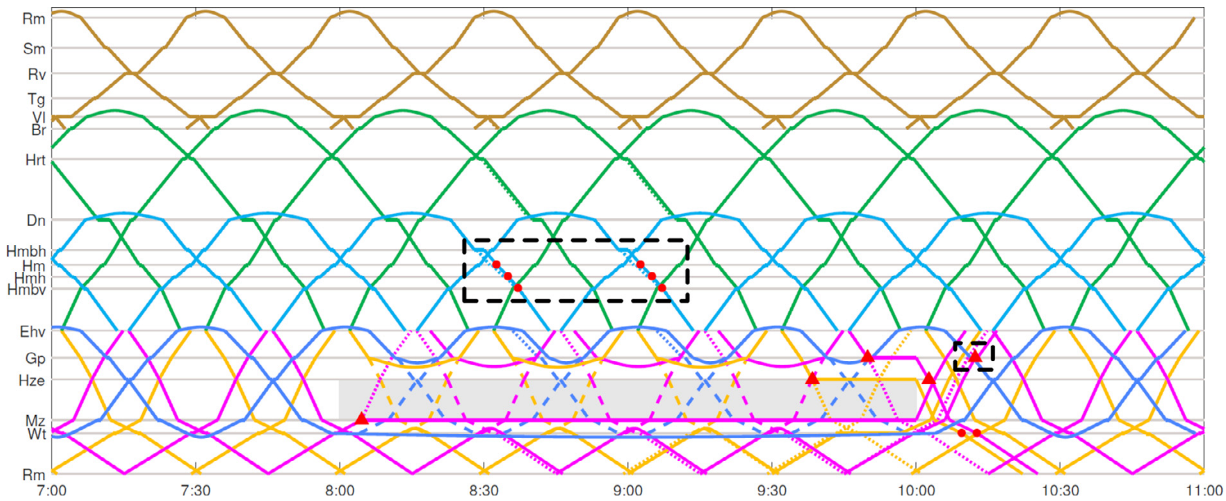


Fig. 8. The optimal passenger-oriented rescheduling solution for case IV: disrupted section Mz-Hze and $\Delta = 10$.

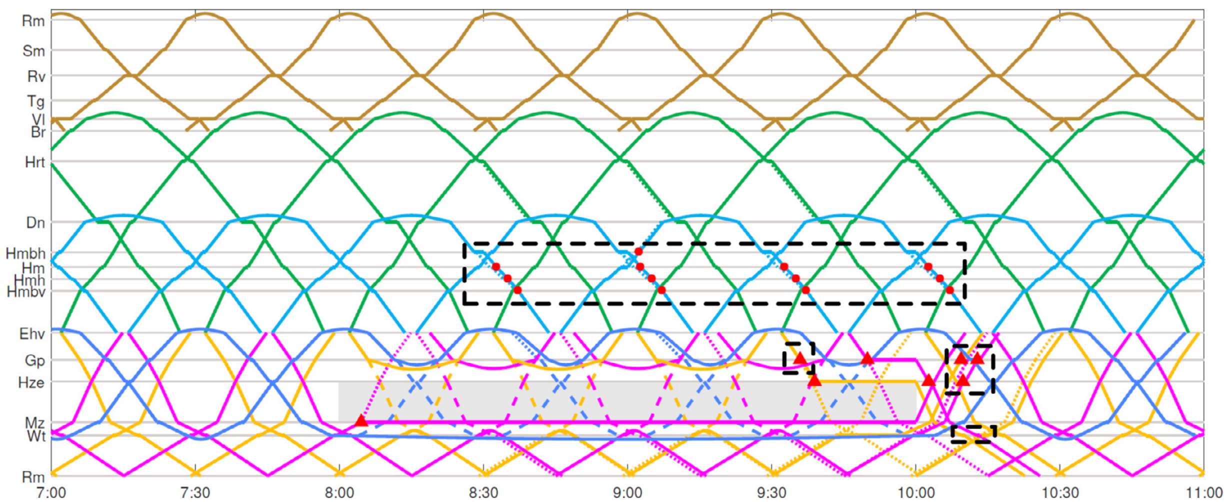


Fig. 9. The optimal passenger-oriented rescheduling solution for case IV: disrupted section Mz-Hze and $\Delta = 30$.

differences of train stopping patterns in each of the passenger-oriented solutions (Fig. 8 or Fig. 9) compared to the operator-oriented solution (Fig. 7) are highlighted by dashed black rectangles. Because station Hze lacks turning facilities, downstream trains from line SPR6400 (in dark blue) and line IC800 (in yellow) were both short-turned at an earlier station Gp. Because station Mz lacks turning facilities for short-turning upstream trains, an upstream train from line IC3500 (in pink) had to be delayed until the disruption ended, and a train from line SPR6400 (in dark blue), which reached its destination (station Wt) around 8:00, had to wait until the disruption ended to operate in opposite direction. These happened in both the operator-oriented solution (Fig. 7) and the passenger-oriented solutions (Fig. 8 and Fig. 9).

Compared to the operator-oriented solution (Fig. 7), the passenger-oriented solution of $\Delta = 10$ (Fig. 8) added 1 more extra stop to an upstream train from line IC3500 (in pink) at station Gp around 10:15, skipped 6 more scheduled stops of two downstream trains from line SPR9600 (in light blue) at stations Hm, Hmh, and Hmbv, and delayed more train services, e.g., the departures of two downstream trains from line SPR9600 (in light blue) at station Hmbh. These departure delays reduced the waiting times of the passengers who arrived at station Hmbh just after the original departure times of these two trains and originally had to board other trains departing later. Due to the delayed departures, the passengers who were on-board these two trains at station Hmbh would experience arrival delays at their destinations so that the model skipped the following stops at stations Hm, Hmh and Hmbv to avoid the destination arrival delays of these passengers. Compared to these on-board passengers, there were much fewer passengers who would board/leave these two trains at station Hm, Hmh or Hmbv, which was another reason why the model decided to skip these stops. Due to the additional stop at station Gp around 10:15 in the passenger-oriented solution of $\Delta = 10$ (Fig. 8), one passenger group (including six passengers) that arrived at station Gp at 10:00 and expected to station Ehv benefitted from earlier boarding by shorter generalized travel times, and another three passenger groups chose not to leave the railways. These three groups include (1) one passenger who arrived at station Rm at 9:00, (2) one passenger who arrived at station Wt at 9:15, and (3) one passenger who arrived at station Rm at 9:45, whose destinations were all station Gp. Given the passenger-oriented solution (Fig. 8), these passengers all took the same train from line IC3500 (in pink), which additionally stopped at station Gp around 10:15. Without this stop (as in the operator-oriented solution shown by (Fig. 7), these passengers would have to take an upstream train to station Ehv first and then transfer to another downstream local train from line SPR6400 (in dark blue) to reach the destination Gp. This would cost much longer than what these passengers would tolerate, which is why they were observed to leave the railways under the operator-oriented solution. Recall that we assume the maximum generalized travel time a passenger is willing to accept under a rescheduled timetable is twice of his/her expected generalized travel time in terms of the planned timetable.

More differences on train stopping patterns from the operator-oriented solution (Fig. 7) were observed in the passenger-oriented solution of $\Delta = 30$ (Fig. 9) due to the increase of Δ, which helped to shorten generalized travel times further. For example, the four more extra stops (highlighted by dashed black rectangles in Fig. 9) helped to shorten the generalized travel times of 31 passengers. Eight of these passengers were observed to leave the railways under the operator-oriented solution (Fig. 7) or the passenger-oriented solution of $\Delta = 10$ (Fig. 8), who however chose to travel by train under the passenger-oriented solution of $\Delta = 30$ (Fig. 9). When $\Delta = 10$ the passenger-oriented solution (Fig. 8) skipped two scheduled stops at station Wt as in the operator-oriented solution (Fig. 7) in order to enable the additional train delay below the current Δ. When increasing Δ to 30 the passenger-oriented solution (Fig. 9) kept these two scheduled stops although leading to more train delay that was acceptable under the current Δ. By keeping these two scheduled stops at station Wt, two more passenger groups chose to not leave the railways compared to either the operator-oriented solution (Fig. 7) or the passenger-oriented solution of $\Delta = 10$ (Fig. 8). These two passenger groups include (1) 15 passengers who arrived at station Ehv at 9:30, and (2) 14 passengers who arrived at station Ehv at 9:45, whose destinations were all station Wt.

These results indicate that the proposed passenger-oriented timetable rescheduling model is able to provide better alternative train services during disruptions with shorter generalized travel times and also helps railway operators to keep more passengers within the railways. By allowing only 10 min additional train delay than the optimal operator-oriented solution, the passenger-oriented model reduced generalized travel times by thousands of minutes, which is a significant improvement to passengers. By allowing more operation deviations from the optimal operator-oriented solution, the passenger-oriented model can reduce generalized travel times further.

Compared to the operator-oriented model, the passenger-oriented model is able to find better rescheduling solutions to passengers while the needed computation times are longer. From Table 10 we know that by using a solver directly an optimal solution can be obtained from the operator-oriented model in seconds, while obtaining an optimal solution from the passenger-oriented model took 370 s in the worst case. With the increase of disruption duration, more passenger groups need to be taken into account, which is why in case I (0.5h disruption) the passenger-oriented model only took 28 s to get an optimal solution, but in case II (1h disruption), case III (1.5h disruption), or case IV (2h disruption) it consumed longer time to generate an optimal solution (see Table 10). Although case III considered a half-hour shorter disruption than case IV, it took a longer computation time than case IV. This is because the disruption durations in both cases are not significantly different and the computation time can be affected by the starting/ending time of a disruption.

7.3. The performance of the AFao algorithm

This section explores the performance of the AFaO algorithm under different parameter settings, including the value of n_{new} (Section 7.3.1), the value of Δ (Section 7.3.2), the disrupted locations (Section 7.3.3), the sorting method of passenger groups G (Section 7.3.4), and the value of μ (Section 7.3.5). How the solution quality evolves over iterations is investigated in Section 7.3.6. Note that except for Section 7.3.4, passenger groups G are sorted in descending order according to the expected total generalized travel times of all passengers in a group in terms of the planned timetable. In that sense, a group g with a larger value of $n_g T_g^{\text{plan}}$ will be handled in an earlier iteration.

7.3.1. The influence of n_{new} on the AFaO algorithm

To solve the passenger-oriented model in a more efficient way, we proposed an AFaO algorithm in Section 6, which is used here to solve the passenger-oriented model for the same cases considered in Table 10. The results from the AFaO algorithm are indicated in Table 13, in which n_{new} refers to the number of passenger groups newly considered in an iteration. I_{need} and I_{finish} refer to the number of required iterations and the number of completed iterations within T^{stop}, respectively. Recall that we set the computation time limit of each iteration to 30 s and the total computation limit to 300 s. Table 13 showed that when n_{new}=10 the AFaO algorithm terminated when reaching the required computation limit in case III or IV, whereas when n_{new}=50 or 100 the required iterations were all completed within the required computation limit in each case. It was observed that with more passenger groups newly considered in an iteration, the total computation time was shorter and the obtained solution was also better. This is because larger n_{new} requires less iterations, at each of which the quality of the solution obtained can also be improved. For each disrupted section the appropriate value of n_{new} to obtain a good solution in short time can be different. For disrupted section Mz-Hze, $n_{\text{new}} = 100$ is better than $n_{\text{new}} = 50$ while we found that for disrupted section Hze-Gp $n_{\text{new}} = 50$ is better than $n_{\text{new}} = 100$. This will be introduced later.

Table 13
Results of applying the AFaO algorithm and the solver to the passenger-oriented model: disrupted section Mz-Hze, and $\Delta = 10$.

Case	AFaO algorithm n_{new}=10					n_{new}=50					n_{new}=100				
	z_p [min]	I_{need}	I_{finish}	Time [sec]	O-gap (z_p) [%]	z_p [min]	I_{need}	I_{finish}	Time [sec]	O-gap (z_p) [%]	z_p [min]	I_{need}	I_{finish}	Time [sec]	O-gap (z_p) [%]
I	112,140	34	34	107	1.23	110,770	7	7	33	0.01	110,758	4	4	29	0.00
II	156,311	48	48	225	1.12	155,408	10	10	70	0.54	154,668	5	5	58	0.06
III	191,078	62	50	300	0.92	189,493	13	13	120	0.09	189,500	7	7	93	0.09
IV	217,262	73	48	300	0.36	217,018	15	15	155	0.25	217,018	8	8	107	0.25
	Solver														
I	110,757	-	-	28	0.00	110,757	-	-	28	0.00	110,757	-	-	28	0.00
II	154,568	-	-	150	0.00	×	-	-	70	-	×	-	-	58	-
III	195,944	-	-	300	3.38	×	-	-	120	-	×	-	-	93	-
IV	216,491	-	-	250	0.00	×	-	-	155	-	×	-	-	107	-

× : no feasible solution

Table 14
The passenger-oriented solutions for case IV under different Δ: disrupted section Mz-Hze.

Δ	Solver z_p [min]	Time [sec]	O-gap (z_p) [%]	AFaO algorithm (n_{new}=100) z_p [min]	I_{need}	I_{finish}	Time [sec]	O-gap (z_p) [%]
0	221,704	213	0.00	221,704	8	8	104	0.00
10	216,481	250	0.00	217,018	8	8	107	0.25
20	213,693	246	0.00	214,010	8	8	110	0.15
30	212,321	472	0.00	212,365	8	8	124	0.02
40	211,331	640	0.00	211,656	8	8	132	0.15
50	210,439	527	0.00	210,986	8	8	131	0.26
60	209,649	834	0.00	210,038	8	8	131	0.19
70	209,175	1,291	0.00	209,369	8	8	128	0.09
80	208,835	1,935	0.00	209,236	8	8	136	0.19
90	208,477	2,212	0.00	208,842	8	8	134	0.17
100	208,129	4,101	0.00	208,312	8	8	135	0.09
	Average	1,156	0.00			Average	125	0.14

For disrupted section Mz-Hze, under whichever setting of n_{new} the optimality gaps of the obtained solutions were all small, among which the worst case was 1.23% and the best case was 0.00%. We use the computation time of the AFaO algorithm as the computation time limit for the solver, and the results are shown in the lower part of Table 13. It is found that when setting $n_{new} = 10$ in most cases the AFaO algorithm took longer times than the solver and the solutions by the AFaO were slightly worse than the ones by the solver. However when setting n_{new} =50 or 100, in most cases the AFaO algorithm took very short time to find optimal or near-optimal solutions, while the solver cannot even find feasible solutions within these times. Besides, the computation times of the AFaO algorithm under n_{new} =50 or 100 were mostly much shorter than the times required by the solver to compute optimal solutions. Hence for our case, setting n_{new} at least to 50 is good to ensure the computation efficiency and the solution quality by the proposed AFaO algorithm.

7.3.2. The influence of Δ on the AFaO algorithm

In addition, we used case IV as an example to investigate the computational efficiency of the AFaO algorithm when allowing larger maximum operation deviation in the passenger-oriented model.

Table 14 shows the results by using a solver directly and the AFaO algorithm to solve the passenger-oriented model under different values of Δ. The time needed to find an optimal solution by a solver directly became longer with the increase of Δ, because a larger solution space needed to be explored. On average, the solver took 1, 156 s to find an optimal solution, which would not be acceptable for real-time application. In contrast, the AFaO algorithm with $n_{new} = 100$ took 125 s on average to find a near-optimal solution. It was observed that the computation time needed by the AFaO algorithm was much less sensitive to the increase of Δ compared to the solver. This indicates that if Δ is increased, the solution space is larger, and thus the proposed algorithm has better performance than the solver. The passenger-oriented solutions of $\Delta = 10$ and $\Delta = 30$ by the AFaO algorithm are shown in Fig. 10 and Fig. 11, respectively. The dashed rectangles highlight the differences on train stopping patterns compared to the operator-oriented solution (Fig. 7). Compared to the optimal passenger-oriented solution (Fig. 8 or Fig. 9), the passenger-oriented solution by the AFaO algorithm (Fig. 10 or Fig. 11) was slightly different on the stopping patterns of trains that were originally planned to run through the disrupted section. This is because the passengers who planned to travel through the disrupted section were handled at later iterations due to

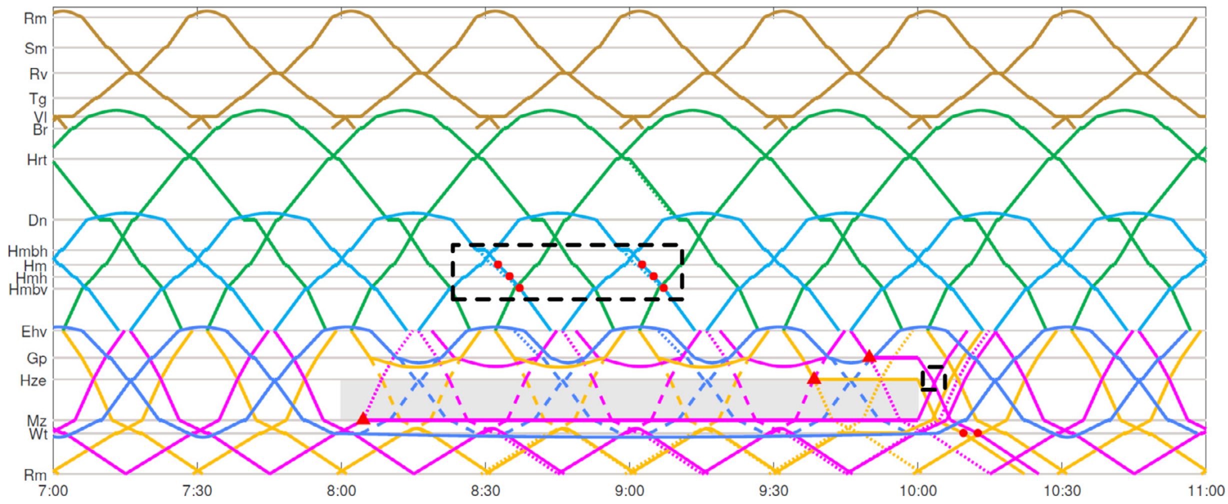


Fig. 10. The sub-optimal passenger-oriented solution by the AFaO algorithm for case IV: disrupted section Mz-Hze and $\Delta = 10$.

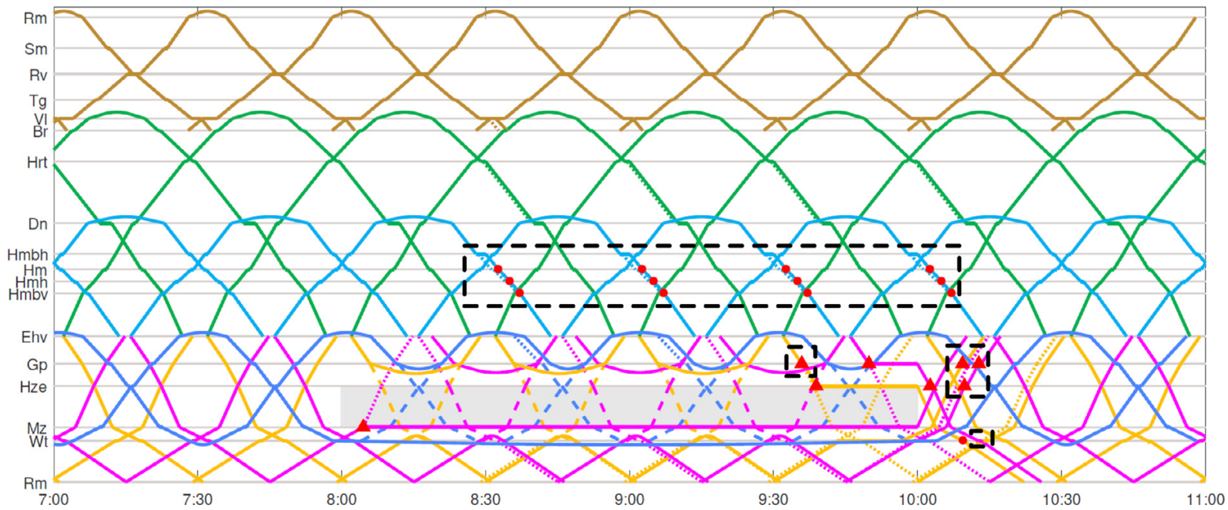


Fig. 11. The sub-optimal passenger-oriented solution by the AFaO algorithm for case IV: disrupted section Mz-Hze and $\Delta = 30$.

their shorter expected generalized travel times in this case, while it was observed that the AFaO algorithm determined the solution mainly according to the needs of passengers handled at earlier solutions.

7.3.3. The influence of disrupted locations on the AFaO algorithm

The previous experiments were carried out on the same disrupted section Mz-Hze. To investigate the performance of the passenger-oriented rescheduling model and the AFaO algorithm on other disrupted locations, we performed experiments to all sections shown in Fig. 4. In each of these experiments, the maximum allowed deviation from the optimal operator-oriented solution Δ is set to 10 in the passenger-oriented model, and the number of passenger groups newly considered in each iteration n_{new} is set to 100 in the AFaO algorithm. Table 15 shows the resulting generalized travel times $\tilde{z}_p$ of the optimal operator-oriented solutions, and the resulting generalized travel time z_p of the optimal passenger-oriented solutions obtained by the solver directly and by the AFaO algorithm. It is observed that an optimal operator-oriented solution was obtained quickly for each disrupted section, but the resulting total generalized travel time is longer than either the one of the optimal passenger-oriented solution or the one of the passenger-oriented solution from the AFaO algorithm. We use ↓ to highlight the decrease in a passenger-oriented solution compared to the corresponding optimal operator-oriented solution. The computation time of generating an optimal passenger-oriented solution by the solver directly varied across disrupted sections. Disrupted section Hrt-Br took the shortest computation time of 41 s, while disrupted section Wt-Mz took the longest computation time of 5, 666 s. The reason is relevant to the number of train lines that were originally scheduled to run through a disrupted section and the starting/ending time of the considered disruption, which both affect the solution space to be explored.

Table 15
Results of different disrupted sections considering case IV.

Operator-oriented				Passenger-oriented: Δ=10								
Disrupted	(solver)			Solver			AFaO algorithm (n_{new}=100)					
section	$\tilde{z}_p$ [min]	Time [sec]	O-gap (z_o) [%]	z_p [min]	Time [sec]	O-gap (z_p) [%]	z_p [min]	I_{need}	I_{finish}	Time [sec]	O-gap (z_p) [%]	$z_p - \tilde{z}_p$ [min]
Rm-Wt	217,814	13	0.00	209,801 ↓	204	0.00	211,040 ↓	8	8	124	0.59	-6,774
Wt-Mz	228,131	12	0.00	214,981 ↓	5,666	0.00	224,217 ↓	8	1	300	4.12	-3,914
Mz-Hze	221,773	12	0.00	216,481 ↓	250	0.00	217,018 ↓	8	8	107	0.25	-4,755
Hze-Gp	223,649	12	0.00	218,086 ↓	534	0.00	220,855 ↓	8	8	132	1.25	-2,794
Gp-Ehv	230,820	12	0.00	225,795 ↓	302	0.00	225,858 ↓	8	8	102	0.03	-4,962
Ehv-Hmbv	241,442	9	0.00	237,066 ↓	388	0.00	237,518 ↓	8	8	102	0.19	-3,924
Hmbv-Hmh	241,508	9	0.00	236,809 ↓	343	0.00	237,316 ↓	8	8	100	0.21	-4,192
Hmh-Hm	241,676	9	0.00	236,854 ↓	353	0.00	237,428 ↓	8	8	99	0.24	-4,248
Hm-Hmbh	231,575	9	0.00	226,125 ↓	253	0.00	227,150 ↓	8	8	74	0.45	-4,425
Hmbh-Dn	233,161	9	0.00	227,983 ↓	252	0.00	228,858 ↓	8	8	85	0.38	-4,303
Dn-Hrt	209,022	9	0.00	201,304 ↓	49	0.00	201,439 ↓	8	8	76	0.07	-7,583
Hrt-Br	205,075	9	0.00	197,694 ↓	41	0.00	198,523 ↓	8	8	69	0.42	-6,552
Br-Vl	203,624	9	0.00	196,495 ↓	57	0.00	197,616 ↓	8	8	79	0.57	-6,008
Vl-Tg	195,050	6	0.00	189,402 ↓	125	0.00	189,402 ↓	8	8	60	0.00	-5,648
Tg-Rv	193,861	6	0.00	188,213 ↓	46	0.00	188,213 ↓	8	8	51	0.00	-5,648
Rv-Sm	193,492	6	0.00	187,640 ↓	67	0.00	187,988 ↓	8	8	52	0.19	-5,504
Sm-Rm	192,653	6	0.00	187,005 ↓	227	0.00	187,005 ↓	8	8	64	0.00	-5,648
Average	217,902	9	0.00	211,631 ↓	539	0.00	212,791 ↓	–	–	99	0.54	-5,075

Table 16
Sorting methods of passenger groups G.

Method No.	Sorting element	Description	Sorting Order
1	$n_g T_g^{\text{plan}}$	The total expected generalized travel times of all passengers in a group g	Descending
2	n_g	The number of passengers in a group g	Descending
3	t_g^{ori}	The arrival time at the origin of group g	Ascending
4	$\tilde{d}_g^{\text{ori}}$	The expected departure time from the origin of group g	Ascending
5	$\tilde{t}_g^{\text{dest}}$	The expected arrival time at the destination of group g	Ascending

From Table 15 we see that in all disrupted sections, the AFaO algorithm found better solutions in terms of generalized travel times than the corresponding operator-oriented solutions. It is observed that the passenger-oriented solution by the AFaO algorithm reduced generalized travel times by thousands of minutes in each disrupted section, which is indicated by $z_p - \tilde{z}_p$. The gap of a solution from the AFaO algorithm to the corresponding optimal passenger-oriented solution was 0.54% on average. The average computation time of obtaining a passenger-oriented solution from the AFaO algorithm was 99 s. In 16 out of 17 disrupted sections, the required iterations were completely finished with at most 132 s in total. An exception was disrupted section Wt-Mz, for which only one iteration was finished and the corresponding computation time reached the required computation time limit: 300 s. This is because disrupted section Wt-Mz was the most difficult section to be solved and thus including 100 new passenger groups in each iteration was still computation-consuming to the passenger-oriented model.

7.3.4. The influence of the sorting method of G *on the AFaO algorithm*

The sequence of passenger groups handled in the AFaO algorithm affects the solution quality. In this section, we introduced five methods of deciding the sequence of passenger groups as shown in Table 16. Each of these methods sorts passenger groups G according to a specific element in a specific order. For example, method 1 sorts passenger groups G in descending order regarding the total expected generalized travel times of all passengers in a group $g \in G$. In that sense, a group g with a larger value of $n_g T_g^{\text{plan}}$ will be handled at an earlier iteration. The values of the sorting elements are estimated according to the expected travel paths of passengers in terms of the planned timetable. The expected travel paths are obtained by the passenger assignment model of Zhu and Goverde (2019a).

The cases solved by the proposed algorithm in Section are all based on the sorting method 1. In this section, we apply the other four sorting methods to the same cases shown in Table 15 of Section 7.3.3, and handle 100 new passenger groups in each iteration of the AFaO algorithm (i.e. $n_{\text{new}} = 100$). The results are shown in Fig. 12, where each subfigure indicates the objective values (generalized travel times) of the passenger-oriented solutions by the proposed algorithm based on the five sorting methods for a specific disrupted section. Note that by whichever sorting method, the obtained passenger-oriented solution is better than the operator-oriented solution in terms of the generalized travel times.

From Fig. 12 we can see that handling passenger groups with larger expected generalized travel times (method 1) or larger passenger volumes (method 2) first resulted in the best solutions in most cases except in disrupted sections Wt-Mz, Mz-Hze, and Hze-Gp. In disrupted section Wt-Mz, handling passenger groups with earlier expected departure times from

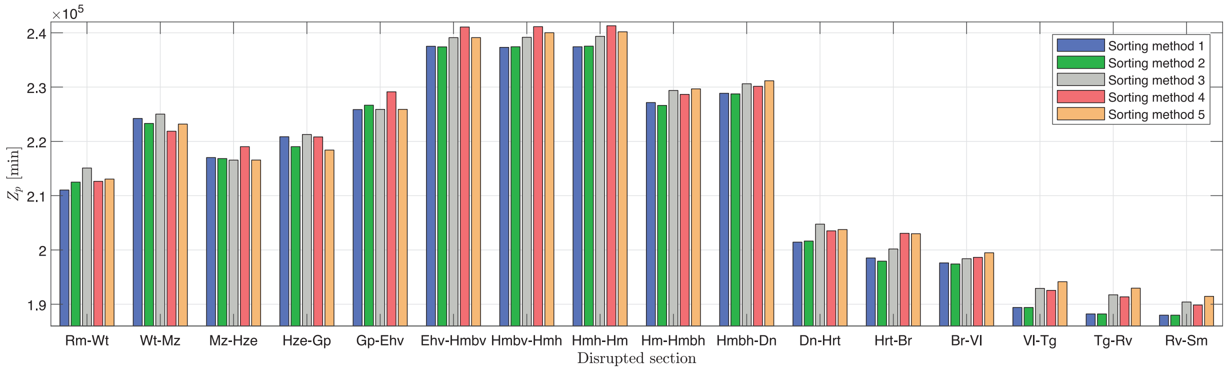


Fig. 12. The objective values (generalize travel times) of the passenger-oriented solutions by the AFaO algorithm based on different sorting methods of G for each disrupted section ($n_{\text{new}} = 100$).

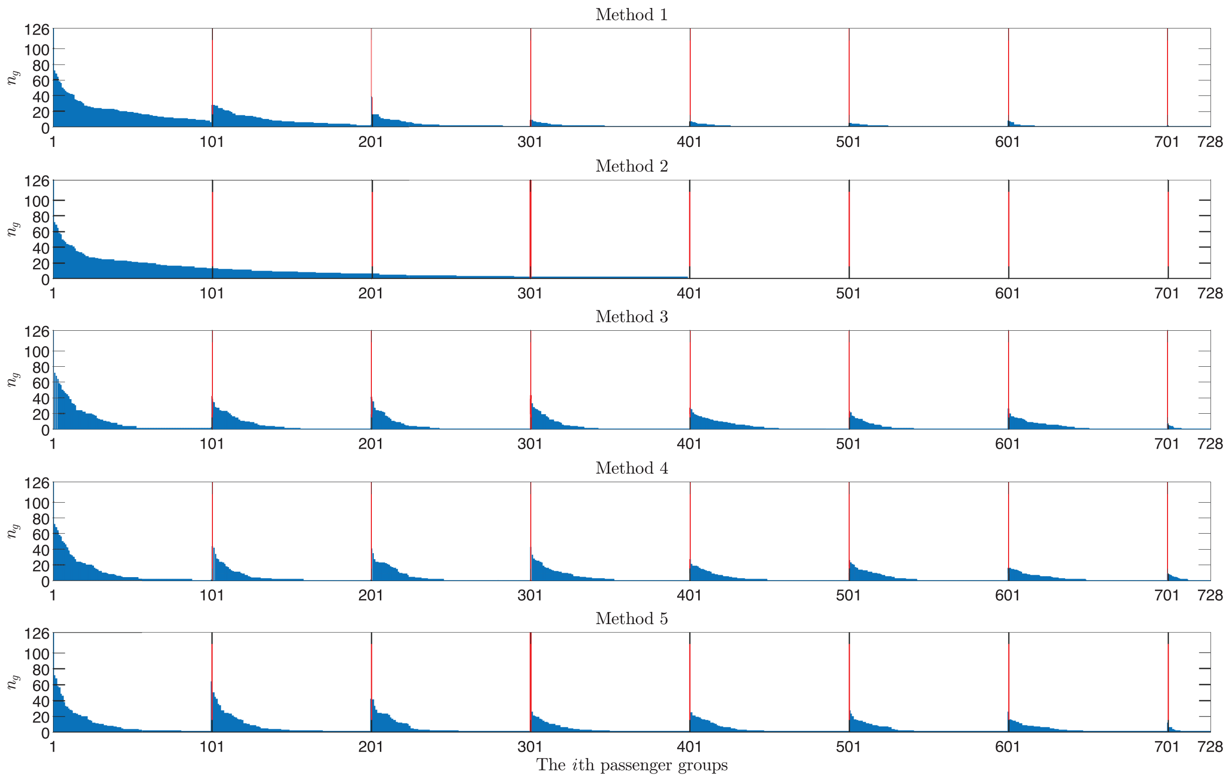


Fig. 13. The size n_g of each group ordered by different sorting methods when $n_{\text{new}} = 100$.

the origins (method 4) first resulted in the best solution. In disrupted sections Mz-Hze, and Hze-Gp, handling passenger groups with earlier expected destination arrival times (method 5) first resulted in the best solution.

These results indicate that the group size (n_g) is an important element to be considered when deciding the sequence of passenger groups handled by the proposed algorithm, although in a few situations other elements might be more important. In general, it is suggested to include larger passenger groups at earlier iterations in the proposed algorithm. Fig. 13 indicates the size n_g of each group ordered by different sorting methods. The largest n_g is 126, while the smallest n_g is 1. The red vertical lines separate the groups newly considered in different iterations ($n_{\text{new}} = 100$). From left to right is iteration 1 to 8. There are 728 groups in total.

7.3.5. The influence of μ on the AFao algorithm

In previous experiments, the value of μ is set to 2, and in this section we increased the value of μ to see how that would affect the results by the AFaO algorithm. Recall that μ determines the maximum generalized travel times accepted by passengers during disruptions. Larger μ means longer acceptable generalized travel times.

Table 17
Results under different settings of μ by the AFaO algorithm: disrupted section Mz-Hze, and $\Delta = 10$.

Case	n_{new}	μ	Disrupted section	#Dropped passengers	Total number of passengers	%Dropped passengers	Time [sec]
IV	100	2	Mz-Hze	1,028	4,357	23.6	107
		2.5		994	4,357	22.8	138
		3		920	4,357	21.1	185
		3.5		866	4,357	19.9	229
		4		604	4,357	13.9	258

Table 18
Results of different disrupted locations under the same setting of μ by the AFaO algorithm: $\Delta = 10$.

Case	n_{new}	μ	Disrupted section	#Dropped passengers	Total number of passengers	%Dropped passengers	Time [sec]
IV	100	2	Mz-Hze	1028	4,357	23.6	107
			Dn-Hrt	508	4,357	11.7	76
			Br-Vl	543	4,357	12.5	79
			Sm-Rm	261	4,357	6.0	64

Table 17 indicates the results under different settings of μ by applying the AFaO algorithm to disrupted section Mz-Hze considering case IV of Table 8. From Table 17 we find that increasing the value of μ does *not* have a large impact on the computational complexity. The computational time grows almost linearly with the increase of μ. The number/percentage of dropped passengers slightly decreases with increasing μ for $\mu \le 3.5$. A big decrease on the number/percentage of dropped passengers was observed for $\mu = 4$. In railway systems in which there are parallel alternative transportation options, the disrupted rail demand can be largely served by other transport means. For example in Melbourne, Australia, half of the disrupted railway demand will be served by bus services implying that 50% passengers will leave the railways (Wang et al., 2014). In the Netherlands, the train operators provide bus shuttle services during railway disruptions. We set $\mu = 2$, under which the resulting percentage of dropped passengers is up to 23.6%. In fact, the number/percentage of passengers leaving the railways is more sensitive to the disrupted locations rather than the value of μ. Table 18 shows that under the same setting of $\mu = 2$ the percentage of passengers leaving the railways ranges from 6% to 23.6% across disrupted sections. More accurate values of μ can be obtained by analysing passenger data during disruptions. This parameter can then be adjusted accordingly.

7.3.6. The quality of a solution obtained at each iteration

The quality of a solution obtained by the AFaO algorithm is determined by the new passenger groups considered at earlier iterations. In this section, we take disrupted sections Wt-Mz, Mz-Hze and Hze-Gp as examples to show how the solution quality evolves over iterations. Passenger groups G are sorted in descending order according to the total expected generalized travel times of all passengers in a group (sorting method 1).

Disrupted section Wt-Mz is the most difficult case as shown in Table 15, which was solved by the proposed algorithm with only one of the 8 required iterations completed within the time limit of 300 s. Fig. 14 shows the iterative solutions of disrupted section Wt-Mz by setting n_{new} to 50 and 100, respectively. For each solution computed in an iteration, we evaluated the resulting generalized travel times of all passengers. The operator-oriented solution was also indicated for comparison. The x-axis refers to the iteration number, and the y-axis refers to generalized travel times. When n_{new}=100, only one iteration was completed due to reaching the total computation time limit of 300 s, while the obtained passenger-oriented solution was still better than the operator-oriented solution. When n_{new}=50, the required 15 iterations were all completed. This indicates that for a disruption case that is difficult to be solved using a smaller value of n_{new} helps to find a better solution quickly. This shows that the proposed algorithm provides a tool to tune the setting of n_{new} to find a good balance between the solution quality and the computation time. It was observed from Fig. 14 that when $n_{\text{new}} = 50$ the passenger-oriented solution by the AFaO algorithm was the same as the operator-oriented solution at the 1st iteration, but was largely improved in the 2nd and 3rd iterations. From the 4th iteration until the final iteration, the passenger-oriented solution was barely improved. This indicates that the quality of the final solution obtained by the AFaO algorithm is mainly determined by earlier iterations. This is because the path choices of passenger groups who have already been considered at an earlier iteration were fixed in the AFaO algorithm at the following iterations where new passenger groups were included but reducing their generalized travel times may increase the ones of earlier considered passenger groups so that very few/none schedule adjustments were made to avoid affecting earlier considered passengers. Recall that in the case study the passenger groups with larger expected generalized travel times are considered at earlier iterations.

We also take disrupted section Mz-Hze and disrupted section Hze-Gp as two more examples to explore the performance of the AFaO algorithm. Fig. show the relevant results, respectively. It is observed that in both disrupted sections, the quality of the passenger-oriented solutions by the AFaO algorithm are mainly determined by earlier iterations when n_{new}=50 or 100, and are all better than the corresponding operator-oriented solutions. In disrupted section Mz-Hze (Fig. 15), a *stable*

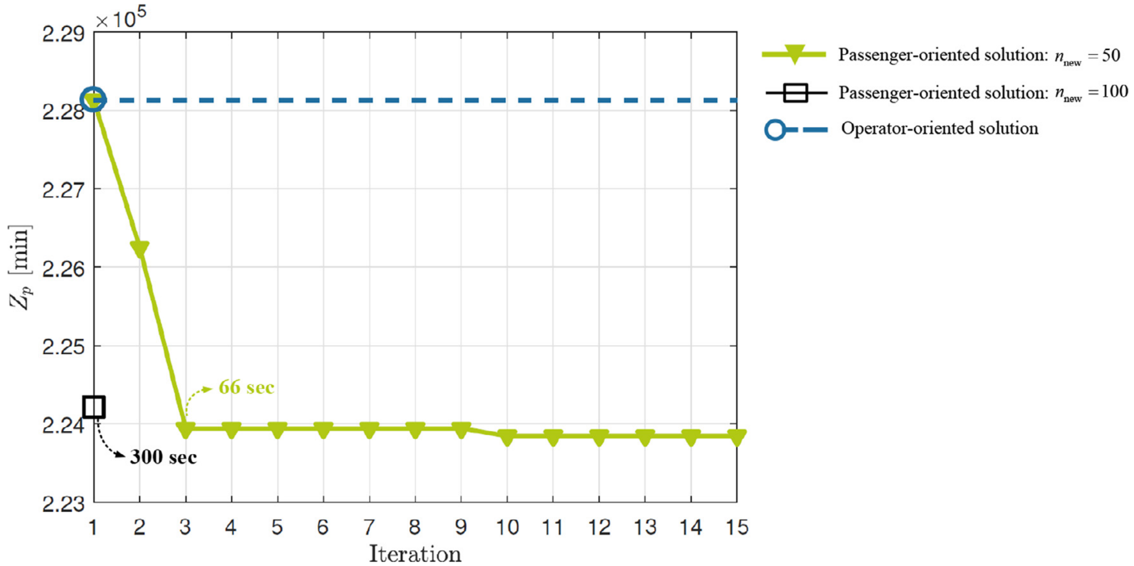


Fig. 14. Results for disrupted section: Wt-Mz (Δ=10).

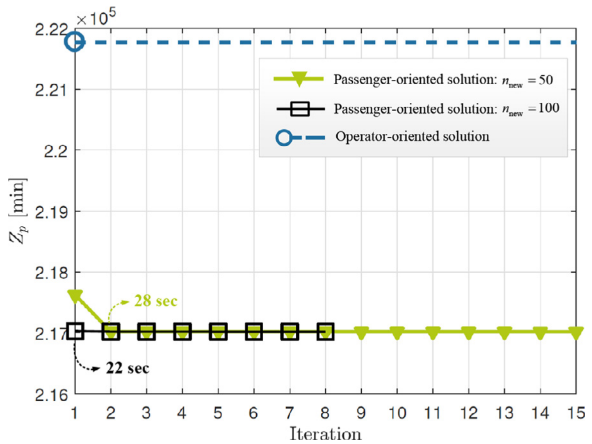


Fig. 15. Results for disrupted section: Mz-Hze (Δ=10).

passenger-oriented solution was obtained after 1 iteration when n_{new}=100, and after 2 iterations when n_{new}=50. Here, we describe a solution as stable when no/few improvements were made on this solution in all following iterations in the AFaO algorithm. The computation time for generating the stable solution was no longer than 30 s when n_{new}=50 or 100. In disrupted section Hze-Gp (Fig. 16), a stable passenger-oriented solution was obtained after 4 iteration when n_{new}=100, and after 3 iterations when n_{new}=50. In these two situations, the computation times for generating the stable solutions were 66 s and 45 s, respectively. It is observed that in disrupted section Hze-Gp the quality of the solution when setting n_{new} to 100 is worse than the quality of the solution when setting n_{new} to 50. The reason is when n_{new}=100 the solution obtained at the 1st iteration was a suboptimal solution, which took 30 s reaching the required computation time limit of an iteration. The relatively poor quality of the 1st solution affects further improvements in the following iterations. Whereas when n_{new}=50, the solution obtained at the 1st iteration was an optimal solution, which helps for further improvements in later iterations.

The results shown in Section 7.3 indicate that the performance of the AFaO algorithm is relevant to the number of passenger groups newly considered at an iteration, the computation time limit required at an iteration, the total computation limit, the disrupted section, and the sorting method of passenger groups G. It is also found that the passenger groups considered at earlier iterations play an important role in determining the quality of the solution finally obtained by the AFaO algorithm. In that sense, the computation time of obtaining a high-quality passenger-oriented solution by the AFaO algorithm can be improved further by only performing a few iterations.

Recall that this paper assumes that trains have unlimited vehicle capacities. To validate the assumption, we checked the number of on-board passengers of each non-cancelled train running through each section, and found that the number was

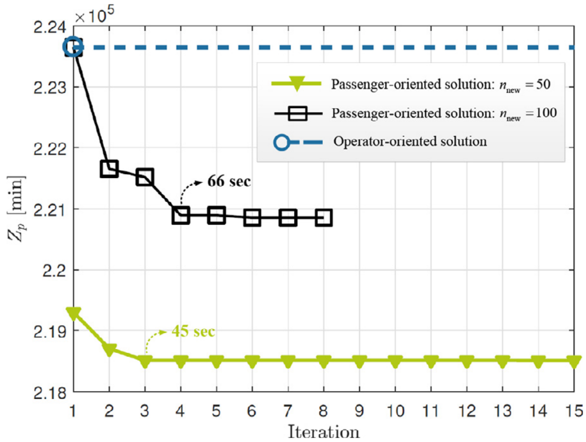


Fig. 16. Results for disrupted section: Hze-Gp (Δ=10).

always below the maximum capacity of the corresponding train. In other words, the capacity of each non-cancelled train was able to handle all passengers who chose to board the train during the disruption.

8. Conclusions and future directions

This paper developed a novel MILP model that integrates timetable rescheduling with passenger reassignment to compute passenger-oriented rescheduled timetables in case of railway disruptions. The objective is minimizing generalized travel times of passengers, which consist of in-vehicle times, waiting times at origin/transfer stations and the number of transfers. Multiple dispatching measures were adopted to adjust the timetable with respect to passenger needs, including re-timing, re-ordering, cancelling, flexible stopping and flexible short-turning trains. An adapted fix-and-optimize algorithm was proposed to solve the model efficiently, by considering restricted passenger groups at each iteration.

The passenger-oriented timetable rescheduling model was applied to a part of the Dutch railways, and compared to an operator-oriented timetable rescheduling model that does not formulate passenger reactions so that the objective is minimizing train cancellations and arrival delays. It was observed that the passenger-oriented model was able to shorten generalized travel times by thousands of minutes with only 10 min more train arrival delay than the optimal operator-oriented solution. With more operation deviations allowed, the passenger-oriented model is able to shorten generalized travel times further. When given a passenger-oriented rescheduling solution, more passengers chose to continue their train travels after the disruption started, compared to an operator-oriented solution for the same disruption case. By the proposed iterative solution method, high-quality rescheduling solutions were obtained by the passenger-oriented model in up to 300 s. It was found that the quality of the final solution obtained by the iterative method is mainly determined by the number of new passenger groups considered at earlier iterations.

In future, we will apply the passenger-oriented model to a larger railway network, by which the computation time will increase further as more train services and passengers will be considered. For this situation, the iterative solution method proposed in this work might also be able to obtain a good solution efficiently as long as representative passenger groups that determine the solution quality can be identified. Then, only these passenger groups need to be handled. During disruptions, trains could become crowded due to detouring passengers whose planned trains were cancelled, and then some passengers would be denied to board specific trains because of lacking capacities. Therefore, we also will take limited vehicle capacity into account to handle both timetable rescheduling and rolling stock rescheduling for providing passengers with more reliable alternative train services in case of railway disruptions. Besides disruption management, the proposed passenger-oriented timetable rescheduling model can be applied to disturbance management by few modifications, which is also promising to be used for improving an existing non-cyclic timetable in terms of generalized travel times. For example, because our model formulates flexible stopping it can be used to determine the planned train stopping patterns of a timetable according to passenger needs.

Acknowledgement

The first author would like to thank the program of China Scholarship Council (No. 201507000056) for the financial support to her PhD research at Delft University of Technology, the Netherlands.

Appendix. Parameters and sets

Table 19
Parameters and sets.

Symbol	Description
$head(a)$	The event which activity a starts from
$tail(a)$	The event which activity a directs to
f_e	Binary parameter with value 1 indicating that arrival event e is a train destination arrival, and 0 otherwise.
r_e	Binary parameter with value 1 indicating that (duplicate) departure event e is a train origin departure, and 0 otherwise.
In_e	Set of activities going in event e
Out_e	Set of activities going out from event e
G	Set of passenger groups
G'	Set of passenger groups that are considered in an iteration in the iterative solution method: $G' \subseteq G$
G_{new}	Set of passenger groups that are newly considered in an iteration in the iterative solution method: $G_{\text{new}} \subseteq G'$
O_g	The origin of passenger group g
D_g	The destination of passenger group g
t_g^{ori}	The origin arrival time of passenger group g
n_g	The number of passengers in passenger group g
T_g^{plan}	The expected generalized travel time of passenger group g in terms of the planned timetable
μT_g^{plan}	The maximum acceptable generalized travel time of passenger group g in terms of a rescheduled timetable: $\mu \geq 1$
$\ell_{\text{trans}}^{\text{min}}$	The minimum transfer time needed at a station
$\ell_{\text{trans}}^{\text{max}}$	The maximum transfer time which a passenger would like to spend at a station
$\ell_{\text{wait}}^{\text{max}}$	The maximum waiting time which a passenger would like to spend at a station
β_{wait}	The multiplier of waiting times perceived by passengers at stations
β_{vehicle}	The multiplier of in-vehicle times perceived by passengers
β_{trans}	The fixed time penalty perceived by passengers on one transfer
t^{stop}	The computation time limit of each iteration in the iterative solution method
T^{stop}	The total computation time limit of the iterative solution method
n_{new}	The number of passenger groups newly considered in each iteration of the iterative solution method
t_{start}	Start time of disruption
t_{end}	End time of disruption
R	The time length required for the planned timetable to be recovered after the disruption ends
D	Maximal allowed delay per event
M	A sufficiently larger number whose value is set to 2880
M^*	A sufficiently larger number whose value is set to $\beta_{\text{wait}} M$

References

Albert, S., Kraus, P., Müller, J.P., Schöbel, A., 2017. Passenger-induced delay propagation: agent-based simulation of passengers in rail networks. In: Simulation Science. Springer, pp. 3–23.

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

Binder, S., Maknoon, Y., Bierlaire, M., 2017. The multi-objective railway timetable rescheduling problem. Transp. Res. Part C 78, 78–94.

Cadarso, L., Marín, Á., Maróti, G., 2013. Recovery of disruptions in rapid transit networks. Transp. Res. Part E 53, 15–33.

Corman, F., D'Ariano, A., Marra, A.D., Pacciarelli, D., Samà, M., 2017. Integrating train scheduling and delay management in real-time railway traffic control. Transp. Res. Part E 105, 213–239.

Dollevoet, T., Huisman, D., Schmidt, M., Schöbel, A., 2012. Delay management with rerouting of passengers. Transp. Sci. 46 (1), 74–89.

Franz, A., Rieck, J., Zimmermann, J., 2019. Fix-and-optimize procedures for solving the long-term unit commitment problem with pumped storages. Ann. Oper. Res. 274 (1–2), 241–265.

Gao, Y., Kroon, L., Schmidt, M., Yang, L., 2016. Rescheduling a metro line in an over-crowded situation after disruptions. Transp. Res. Part B 93, 425–449.

Ghaemi, N., Cats, O., Goverde, R.M.P., 2017. A microscopic model for optimal train short-turnings during complete blockages. Transp. Res. Part B 105, 423–437.

Ghaemi, N., Cats, O., Goverde, R.M.P., 2017. Railway disruption management challenges and possible solution directions. Public Transp. 9 (1–2), 343–364.

Ghaemi, N., Cats, O., Goverde, R.M.P., 2018. Macroscopic multiple-station short-turning model in case of complete railway blockages. Transp. Res. Part C 89, 113–132.

Van der Hurk, E., Kroon, L., Maróti, G., 2018. Passenger advice and rolling stock rescheduling under uncertainty for disruption management. Transp. Sci. 52 (6), 1391–1411.

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: Proceedings of the European Transport Conference, Glasgow, 8–10 October 2012.

Kroon, L., Maróti, G., Nielsen, L., 2014. Rescheduling of railway rolling stock with dynamic passenger flows. Transp. Sci. 49 (2), 165–184.

Lang, J.C., Shen, Z.-J.M., 2011. Fix-and-optimize heuristics for capacitated lot-sizing with sequence-dependent setups and substitutions. Eur. J. Oper. Res. 214 (3), 595–605.

Louwerse, I., Huisman, D., 2014. Adjusting a railway timetable in case of partial or complete blockades. Eur. J. Oper. Res. 235 (3), 583–593.

Meng, L., Zhou, X., 2011. Robust single-track train dispatching model under a dynamic and stochastic environment: a scenario-based rolling horizon solution approach. Transp. Res. Part B 45 (7), 1080–1102.

Sahling, F., Buschkühl, L., Tempelmeier, H., Helber, S., 2009. Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. Comput. Oper. Res. 36 (9), 2546–2553.

Schöbel, A., 2001. A model for the delay management problem based on mixed-integer-programming. Electron. Notes Theor. Comput. Sci. 50 (1), 1–10.

Veelenturf, L.P., Kidd, M.P., Cacchiani, V., Kroon, L.G., Toth, P., 2015. A railway timetable rescheduling approach for handling large-scale disruptions. Transp. Sci. 50 (3), 841–862.

Veelenturf, L.P., Kroon, L.G., Maróti, G., 2017. Passenger oriented railway disruption management by adapting timetables and rolling stock schedules. Transp. Res. Part C 80, 133–147.

Wang, Y., Guo, J., Currie, G., Ceder, A., Dong, W., Pender, B., 2014. Bus bridging disruption in rail services with frustrated and impatient passengers. IEEE Trans. Intell. Transp. Syst. 15 (5).
Wardman, M., 2004. Public transport values of time. Transp. Policy (Oxf) 11 (4), 363–377.
Zhan, S., Kroon, L.G., Veelenturf, L.P., Wagenaar, J.C., 2015. Real-time high-speed train rescheduling in case of a complete blockage. Transp. Res. Part B 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. Transp. Res. Part E 95, 32–61.
Zhu, Y., Goverde, R.M.P., 2017. System-based vulnerability measures for railway systems. In: The 7th International Conference on Railway Operations Modelling and Analysis (IAROR): RailLille.
Zhu, Y., Goverde, R.M.P., 2019. Dynamic passenger assignment model for major railway disruptions considering information interventions. Netw. Spatial Econ. 19 (4), 1249–1279.
Zhu, Y., Goverde, R.M.P., 2019. Dynamic railway timetable rescheduling for multiple connected disruptions. Transp. Res. Part C (Submitted for publication).
Zhu, Y., Goverde, R.M.P., 2019. Railway timetable rescheduling with flexible stopping and flexible short-turning during disruptions. Transp. Res. Part B 123, 149–181.