



Delft University of Technology

Dynamic Passenger Assignment for Major Railway Disruptions Considering Information Interventions

Zhu, Yongqiu; Goverde, Rob

DOI

[10.1007/s11067-019-09467-4](https://doi.org/10.1007/s11067-019-09467-4)

Publication date

2019

Document Version

Final published version

Published in

Networks and Spatial Economics

Citation (APA)

Zhu, Y., & Goverde, R. (2019). Dynamic Passenger Assignment for Major Railway Disruptions Considering Information Interventions. *Networks and Spatial Economics*, 19(4), 1249-1279.
<https://doi.org/10.1007/s11067-019-09467-4>

Important note

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

Copyright

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

Takedown policy

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



Dynamic Passenger Assignment for Major Railway Disruptions Considering Information Interventions

Yongqiu Zhu¹  · Rob M. P. Goverde¹

Published online: 30 April 2019
© The Author(s) 2019

Abstract

Passenger assignment models for major disruptions that require trains to be cancelled/short-turned in railway systems are rarely considered in literature, although these models could make a significant contribution to passenger-oriented disruption timetable design/rescheduling. This paper proposes a dynamic passenger assignment model, where the passengers who start travelling before, during and after the disruption are all considered. The model ensures that on-board passengers are given priority over waiting passengers, and waiting passengers are boarding under the first-come-first-serve rule. Moreover, the model allows information interventions by publishing information about service variations and train congestion at different locations with the aim of distributing passengers wisely to achieve less travel time increase due to the disruption. Discrete event simulation is adopted to implement the model, where loading/unloading procedures are realized and passengers re-plan their paths based on the information they receive. The model tracks individual travels, which helps to evaluate a disruption timetable in a passenger-oriented way.

Keywords Major disruptions · Railway systems · Dynamic assignment · Information intervention

1 Introduction

Unexpected events affect railway operations in everyday life, which are either small service perturbations called *disturbances* or relatively large incidents called *disruptions*. During disturbances, train services will be delayed, but not cancelled/short-

✉ Yongqiu Zhu
y.zhu-5@tudelft.nl

Rob M. P. Goverde
r.m.p.goverde@tudelft.nl

¹ Department of Transport and Planning, Delft University of Technology,
Stevinweg 1, 2628 CN Delft, The Netherlands

turned which however is necessary during disruptions. Due to the complexity of handling disruptions, contingency plans are designed beforehand for different disruption scenarios. When a disruption happens, the corresponding contingency plan is selected, and possibly modified by traffic controllers in terms of the specific condition (Ghaemi et al. 2017b). However, in either the design or modification procedure, passengers who should have been put first, are as yet not incorporated directly, because traffic controllers are unable to anticipate the passenger flows over the network. As a result, many alternatives for passenger reroutings are not considered, and thus passenger travel experiences during disruptions are usually less than satisfactory.

To support passenger-oriented train service adjustments, it is necessary to have a passenger assignment model that can anticipate the distribution of passengers. Based on the model, whether a timetable is passenger-friendly or not can be evaluated, and further how to adjust the timetable in a passenger-friendly way can be guided. Until now, passenger assignment models are mostly proposed for planning purposes or disturbance management (generally regarded as *delay management*), where services are considered to be reliable or with minor perturbations. When major disruptions like complete track blockages occur, multiple dispatching measures, e.g. retiming, reordering, cancelling and short-tuning trains, are commonly applied, which result in delayed trains, changed train orders, completely cancelled trains and short-turned trains (Ghaemi et al. 2017a). As a result, the train services available during disruptions are rather different from the ones on normal days, thus leading to rather different path options to passengers. For passenger assignment models during disruptions, it is necessary to formulate the major service variations properly and model passenger responses to such major service variations accurately. Therefore, this paper proposes a dynamic passenger assignment model taking major service variations, vehicle capacity, and time-dependent passenger all into account. A preliminary version of the model can be found in Zhu and Goverde (2017), which is improved by introducing a new network formulation and information interventions for altering passenger behaviour in this paper.

This paper considers passengers' en-route travel decisions rather than passengers' pre-trip travel decisions. This means that passengers are assumed to have planned paths in mind before they actually arrive at the origin stations, however, possibly they have to re-plan their paths due to major service variations, denied boardings or train congestion. Such an assumption is justified, since nowadays passengers can rely on various travel-planner applications or the official websites of operator companies to find their preferred paths. This is particularly true for passengers who have a clear travel purpose (e.g. commuters). Thus, once disruptions occur, passengers would make en-route travel decisions by comparing the alternative paths during disruptions with their planned paths.

Passenger attitudes towards path alternatives during disruptions could be different from the ones on normal days. For example, due to reduced operation frequency during disruptions, passengers may be willing to spend more waiting times at origin/transfer stations than usual. Considering this, a new method is proposed

to formulate the network with less arcs, which ensures all paths that could be chosen by passengers to be fully covered. The formulated network is a directed acyclic graph (DAG) with passenger perceived times on arcs, based on which the optimal paths perceived by passengers can be searched using efficient shortest path algorithms.

Path alternatives can be different if passengers re-plan paths at different locations and times. This paper tracks the location of each passenger who starts travelling before, during, or after the disruption, and decides when and where he/she re-plans the path based on the information received. Information interventions are considered by delivering two kinds of information, service variations and train congestion, separately at different locations. Usually, the congestion effect is considered as the increase in travel times perceived by passengers (Cats et al. 2016; Larrain and Muñoz 2008). Instead, this paper aims to avoid travel time increase due to denied boarding, by using congestion information to affect passenger behavior in the following way. Imagine that a train is highly congested when departing from a stop, and there are still many passengers wishing to board this train at its next stop. It is possible that the train is unable to handle all these passengers. Thus, only some of them can board the train successfully, while the others have to be denied. If there must be some passengers being denied for boarding a train, avoiding them to choose the train may help them find better alternative paths compared to the ones they can find after being denied. Considering this situation, if a train is potentially unable to handle all passenger demand at its next stop, part of these passengers are notified with the congestion information in order to encourage them to choose another train, while the other part of these passengers are kept unaware of such information to ensure they will stay with their choice for this train.

The key contributions of this work are summarized as follows:

- Proposing a new schedule-based passenger assignment model during major disruptions.
- Developing a new network formulation to formulate the timetable as a directed acyclic graph (DAG) with passenger perceived times on arcs.
- Taking time-dependent passenger demand, service variations, and vehicle capacity constraints all into account.
- Formulating passenger responses towards major service variations, like short-turned or cancelled trains.
- Using information interventions to influence passenger behaviour.
- Dealing with passengers who start travelling before, during and after the disruption.

The remainder of this paper is organized as follows. Section 2 gives an overview of the relevant work. Section 3 explains the network modelling approach. In Section 4, the proposed dynamic passenger assignment framework is shown, followed by the explanation of the main parts in the framework. Next, the time complexities of the proposed algorithms are analysed in Section 5. Finally in Section 6, a case study of a complete open track blockage in part of the Dutch railway network is performed.

2 Literature Review

Passenger assignment models for transit systems are typically classified into schedule-based and frequency-based (Gentile and Noekel 2016), differing in whether passengers make route choices in terms of the timetable that indicates the departure/arrival time of each train at each station. In general, frequency-based models are suitable for such transit systems where the operations are so frequent that passengers can be assumed to board the first train when waiting at a station. While in railway systems where the operation frequency is relatively low, schedule-based models are commonly used, like Binder et al. (2017a) and Rückert et al. (2017).

Some assignment models are proposed for planning purposes, for example, identifying the phenomenon of macroscopic congestion of a proposed transit system. In these models, services are assumed to be constant or affected by minor perturbations that do not require dispatching measures to be applied. For instance, Khani et al. (2015) propose three path searching algorithms to make the assignment model perform efficiently on large-scale transit networks, by assuming that the operation is reliable and vehicle capacity is infinite. With limited vehicle capacity considered, Poon et al. (2004), Hamdouch and Lawphongpanich (2008), and Binder et al. (2017a) explore the interactions between the supply and the demand over time, which differ in the used priority rules for passenger boardings while share the assumption of trains operating precisely on schedule. In practice, service variations cannot be fully avoided. Thus, Nuzzolo et al. (2001), Hamdouch et al. (2014), and Cats et al. (2016) take service variations into account, and describe the variations as irregularities of train dwell and running times that are thought to be relevant to the passenger loadings of the corresponding trains. The considered train delays do not need timetable rescheduling, which means that train orders remain unchanged and no trains are cancelled or short-turned.

When train delays cannot be absorbed completely by the time supplements reserved in the timetable, timetable rescheduling becomes necessary. A typical question under such a case is that whether a train should wait for a delayed feeder train or better depart on time (*wait-depart decision*). This problem is generally regarded as delay management, where the relevant work mainly focuses on the optimization and thus the formulation corresponding to the passenger assignment is usually simplified by some assumptions. For example, Schöbel (2001) assumes that once passengers miss a transfer connection, they would wait for a complete cycle time to catch the next connection. Kanai et al. (2011), Dollevoet et al. (2012), Sato et al. (2013), and Corman et al. (2017) consider the alternative choices that passengers might have, where the capacities of vehicles are assumed to be infinite. While most papers consider the train delays as known input to the optimization, Rückert et al. (2017) observe the train delays in real time, and predict the passenger flows due to any possible wait-depart decisions to help the dispatchers make informed decisions. In these papers, train orders can be changed, but no trains are delayed significantly or cancelled/short-turned, which however take place during disruptions.

A few papers consider the passenger assignment during disruptions. Cats and Jenelius (2014) focus on disruptions that result in trains delayed significantly. The considered case is that the tracks between two stations are totally blocked for 30

minutes, and trains queue at the station before the blocked tracks during the disruption period. When the disruption ends, all these trains are again allowed to continue the following operations, assuming that all on-board passengers in these delayed trains are unable to alight from the trains at the holding stations. For a long-duration disruption that lasts for one hour or even more, it is unlikely to hold trains at stations, but more likely to short-turn them. In such a case, on-board passengers must alight from the trains, since the trains can no longer reach their expected destinations. Binder et al. (2017b) formulate the passenger assignment as a multi-commodity problem and integrate it with the rescheduling together constituting a passenger-oriented timetable rescheduling model for disruptions. The considered demand is the passengers who start travelling during the disruption. An assumption is implicitly made that all passengers collaborate together to achieve the system optimum. In the real world, passengers may intend to reduce their personal inconvenience without considering and of course incapable of considering the impacts of their choices on the system optimum. Thus, treating passengers as rational actors is necessary, which can be implemented by introducing priority rules for passenger boardings.

The literature does not consider passengers' en-route travel decisions during *major* disruptions for which cancelling/short-turning trains are necessary. This paper fills the gap by proposing a schedule-based passenger assignment model to formulate the changes of passenger responses from normal situations to during disruptions.

The model is based on three assumptions, which are also used in Cats and Jenelius (2014) and Binder et al. (2017b). First, at the beginning of a disruption, the exact disruption end time is known, which will not be extended or shortened. This assumption can be relaxed by embedding the proposed model into an iterative framework where at each iteration the disruption end time is updated and the model is performed again based on the renewed disruption information and the corresponding disruption timetable. The second assumption is that for the railway operators, the disruption timetable is available directly at the beginning of the disruption. This is possible when applying a real-time optimization model (e.g. Ghaemi et al. 2017a) to compute the disruption timetable. The third assumption is that the passenger demand during disruptions is the same as on normal days. This assumption is relaxed due to setting the maximum acceptable delay of the re-planned path. In the model, a passenger can drop the railways if the delay due to the re-planned path is not acceptable. Thus, although a passenger is assumed to come to the railway origin station as planned, he/she could immediately leave if the planned path is inapplicable and the minimal delay across the current alternative paths provided by the railways exceeds the maximum acceptable delay. Such an immediate leaving is actually equal to not coming to the railways.

3 Event-Activity Network

A transit assignment model depends on the network formulation that enables travel path generation for passengers. This paper proposes a new approach to formulate the train services as a weighted DAG based on which the optimal paths perceived by passengers can be quickly searched. The characteristics of railway timetables (e.g.

overtakings) and the fact that passengers might choose unusual paths (e.g. the ones with long waiting/transfer times at stations) during disruptions, are all considered in the proposed network formulation. As events are used to represent nodes and activities are used to represent arcs, the formulated network is called an event-activity network. In the following, different kinds of events and activities that are necessary to formulate the network are introduced, as well as the passenger preferred weights on the activities.

3.1 Events

There are four types of events in the formulated network. They are arrival events, departure events, duplicate departure events and exit events, which constitute the sets E_{arr} , E_{dep} , E_{ddep} and E_{exit} , respectively. Therefore, the set of events is

$$E = E_{\text{arr}} \cup E_{\text{dep}} \cup E_{\text{ddep}} \cup E_{\text{exit}}. \quad (1)$$

For each event $e \in E$, the attribute st_e that indicates the corresponding station of e is assigned. Additionally for each event $e \in E_{\text{arr}}$ or E_{dep} , two more attributes tr_e and π_e are assigned, which refer to the corresponding train and occurrence time of e , respectively. An event $e \in E_{\text{ddep}}$ is the duplicate of a specific departure event with the exactly same attributes that the departure event has. One and only one duplicate is created for each departure event. The reason of creating duplicate departure event is to construct waiting and transfer activities, which is explained in more detail in Section 3.2. The notation of event attributes is described in Table 1 while the attributes of different events are shown in Table 2.

3.2 Activities

There are five types of activities in the formulated network. They are running activities, dwell activities, wait activities, transfer activities and exit activities, which constitute the sets A_{run} , A_{dwell} , A_{wait} , A_{trans} and A_{exit} , respectively. In addition, A_{wait} consists of two sub-sets that are A_{ddw} and A_{adw} , which correspond to the wait activities between duplicate departure events and the wait activities between arrival events and duplicate departure events. Namely,

$$A_{\text{wait}} = A_{\text{ddw}} \cup A_{\text{adw}}. \quad (2)$$

Therefore, the set of activities is

$$A = A_{\text{run}} \cup A_{\text{dwell}} \cup A_{\text{ddw}} \cup A_{\text{adw}} \cup A_{\text{trans}} \cup A_{\text{exit}}. \quad (3)$$

Table 1 Notation of event attributes

Symbol	Description
st_e	The station of event e
tr_e	The train of event e
π_e	The occurrence time of event e

Table 2 Attributes of different events

Event	Attributes
Arrival event: $e \in E_{arr}$	(st_e, tr_e, π_e)
Departure event: $e \in E_{dep}$	(st_e, tr_e, π_e)
Duplicate departure event: $e \in E_{ddep}$	(st_e, tr_e, π_e)
Exit event: $e \in E_{exit}$	st_e

Running activities enable passengers travelling from one station to another:

$$A_{run} = \{(e, e') \mid e \in E_{dep}, e' \in E_{arr}, tr_e = tr_{e'} \text{ and } st_e \text{ is upstream neighbouring to } st_{e'}\}. \quad (4)$$

Dwell activities enable passengers dwelling at the station in a train:

$$A_{dwell} = \{(e, e') \mid e \in E_{arr}, e' \in E_{dep}, tr_e = tr_{e'} \text{ and } st_e = st_{e'}\}. \quad (5)$$

Wait activities and transfer activities together enable passengers waiting to board trains at origins or transferring from one train to another at other stations:

$$A_{ddw} = \left\{ (e, e') \mid e \in E_{ddep} \text{ and } e' = \arg \min \{ \pi_{e'} \mid \pi_{e'} > \pi_e : e' \in E_{ddep}, tr_{e'} \neq tr_e, st_{e'} = st_e \} \right\}, \quad (6)$$

$$A_{adw} = \left\{ (e, e') \mid e \in E_{arr} \text{ and } e' = \arg \min \{ \pi_{e'} \mid \pi_{e'} > \pi_e : e' \in E_{ddep}, tr_{e'} \neq tr_e, st_{e'} = st_e \} \right\}, \quad (7)$$

$$A_{trans} = \left\{ (e, e') \mid e \in E_{ddep}, e' \in E_{dep}, tr_e = tr_{e'}, st_e = st_{e'} \text{ and } \pi_e = \pi_{e'} \right\}. \quad (8)$$

Here, Eq. 6 means that each duplicate departure event is linked to the next time-adjacent duplicate departure event that is at the same station but for another train. Similarly, Eq. 7 means that each arrival event is linked to the next time-adjacent duplicate departure event that is at the same station but for another train. Finally, Eq. 8 means that each duplicate departure event is linked to its original departure event.

Exit activities enable passengers to leave the railway system once arriving at the destinations:

$$A_{exit} = \{(e, e') \mid e \in E_{arr}, e' \in E_{exit}, st_e = st_{e'}\}. \quad (9)$$

In Fig. 1, the formulated event-activity network is shown for an example with four stations (i.e. A, B, C and D) and three trains numbered 1 to 3. The attributes corresponding to each event are enclosed in an ellipse, rectangle or circle that refer to an arrival/departure event, duplicate departure event or exit event, respectively. For instance, an ellipse with $(dep, 1, B)$ represents the departure event of train 1 at station B. A path is represented by a series of time-ordered events. For example, one of the paths available for a passenger who arrives at station A after time t_1 but before time t_2 and wishes to travel to station D is: $(ddep, 2, A) \rightarrow (dep, 2, A) \rightarrow (arr, 2, B) \rightarrow$

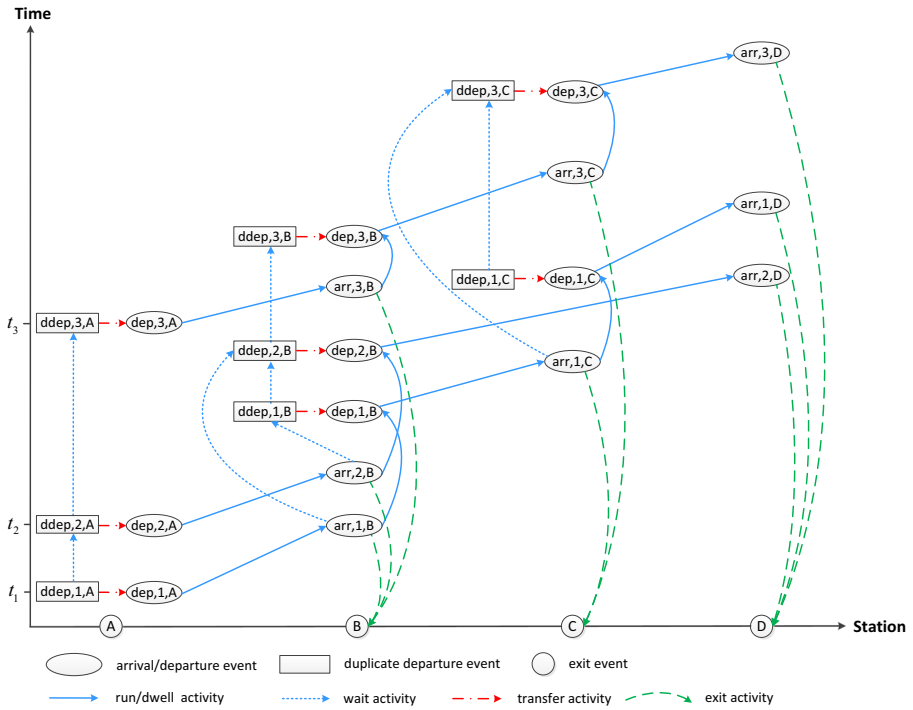


Fig. 1 Event-activity network

$(ddep,1,B) \rightarrow (dep,1,B) \rightarrow (arr,1,C) \rightarrow (dep,1,C) \rightarrow (arr,1,D)$. This path means that the passenger boards train 2 at station A, but transfers to train 1 at station B and stays in this train until the destination (i.e. station D).

3.3 Weights of Activities

Usually, paths are perceived differently by passengers due to the path attributes like waiting time at the origin (t_{origin}), in-vehicle time (t_{vehicle}), waiting time at a transfer station (t_{trans}), and number of transfers (n_{trans}). A utility function is used to quantify the utility of each path by giving different weights on the path attributes. In this paper, the utility of a path r is quantified as:

$$U_r = \beta_1 t_{\text{vehicle}}^r + \beta_2 (t_{\text{origin}}^r + t_{\text{trans}}^r) + \beta_3 n_{\text{trans}}^r, \quad (10)$$

where β_1 , β_2 , β_3 are the weights of the corresponding attributes of path r . Here, the values of β_1 , β_2 and β_3 are set as 1, 2 (Wardman 2004) and 10 (de Keizer et al. 2012) for each minute, respectively.

In this paper, the path utility calculation is realized in the procedure of path searching. This means that once a path is generated by a path search algorithm (e.g. shortest path algorithm, k -shortest path algorithm, etc.), the *distance* of this path is actually

the utility of this path. For this purpose, different weights are assigned to different activities, as follows.

- For each activity $a = (e, e') \in A_{\text{run}} \cup A_{\text{dwell}}$, the weight of a is set as $\beta_1(\pi_{e'} - \pi_e)$.
- For each activity $a = (e, e') \in A_{\text{ddw}} \cup A_{\text{adw}}$, the weight of a is set as $\beta_2(\pi_{e'} - \pi_e)$.
- For each passenger p , the weight of a transfer activity could be different depending on where the passenger started travelling. For an activity $a = (e, e') \in A_{\text{trans}}$, the weight of a is set as zero if $st_e = o_p$ (o_p is the origin of p). Otherwise (i.e. $st_e \neq o_p$), the weight of a is set as a fixed value β_3 .
- For each activity $a \in A_{\text{exit}}$, the weight of a is set to the same positive value, since it is not used to distinguish paths.

The weights of all activities are contained in the set W . Thus, the formulated event-activity network is

$$G = (E, A, W). \quad (11)$$

3.4 Searching the Optimal Path Perceived by Passengers

Let a passenger p have the attributes (o_p, d_p, t_p^o) referring to the origin, destination and actual arrival time at the origin, respectively. To search the optimal path perceived by p for the travel from o_p to d_p , a pair of source and sink nodes should be given. Here, the sink node v is defined as

$$v = \{e \in E_{\text{exit}} \mid st_e = d_p\}, \quad (12)$$

and the source node u is defined as

$$u = \arg \min \left\{ \pi_e \mid \pi_e \geq t_p^o : e \in E_{\text{ddep}}, st_e = o_p \right\}, \quad (13)$$

which means that the source node u is set as the duplicate departure event e at the origin station o_p , of which the occurring time π_e is closest to the passenger's arrival time at the origin t_p^o . Note that defining the source node this way takes the passenger's choice about the boarding train at the origin into account. For example in Fig. 1, suppose $(depp, 1, A)$ is chosen as the source node for a passenger who plans to travel from station A to station D. Then, the passenger could take train 1 as the first boarding train, or also could wait a bit longer to take train 2 as the first boarding train.

With the assigned pair of source and sink node, the shortest path in utility can be searched, by performing a shortest path algorithm on the formulated event-activity network G , which by construction is actually a directed acyclic graph (DAG) with positive arc weights. Such a shortest path algorithm topologically sorts the nodes of DAG in passenger perceived times, thus making the predecessor node of an edge always appear before the successor node of the edge in a linear ordering (Cormen et al. 2009). Using the topological order, the shortest path is found in time complexity $O(A + E)$.

Here, the optimal path perceived by a passenger p is represented by r_p , which is first searched by the shortest path algorithm and then processed by excluding the duplicate departure events and the exit event. In other words, r_p only consists of the events that directly serve the path. Based on r_p , the departure/arrival events of p

are extracted further, which correspond to the boarding/alighting actions. Here, B_p represents the set of departure events that correspond to the boarding actions of p at origin and transfer stations (if any), and L_p represents the set of arrival events that correspond to the alighting actions at transfer stations (if any) and destination.

Note that the way of deciding the source node in Eq. 13 is only for the passengers who are at the origins before travelling (and have not been denied for boarding). For the passengers who have already started travelling within trains, at transfer stations, or at the stations where they are forced to get off due to cancelled services, the ways of choosing the source nodes for searching paths are different, which are explained in Section 4.3.4.

4 Dynamic Assignment Model

The framework of dynamic passenger assignment during disruptions consists of three parts, as shown in Fig. 2.

- Part I assigns each passenger to a planned path based on the original timetable.

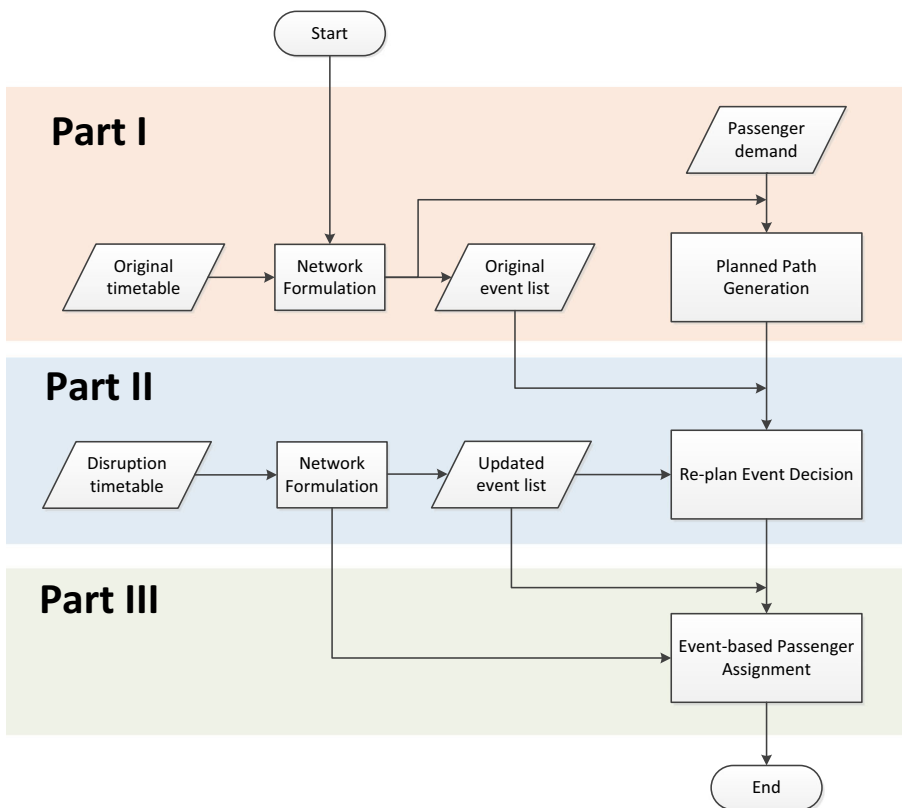


Fig. 2 Framework of the dynamic passenger assignment model during major disruptions

- Part II decides which passengers are affected under the disruption timetable due to delayed/cancelled services, and also decides when these affected passengers would re-plan the paths considering different locations of publishing service variations.
- Part III simulates passenger loading and unloading procedures and also the path re-plannings of passengers because of service variations, denied boardings, or train congestion.

In what follows, the three parts are introduced successively. The used notation is described in Table 3.

4.1 Passenger Planned Path Assignment (Part I)

In part I, the original timetable is formulated as an event-activity network G_{plan} where the arrival events and departure events form the original ordered event list $E_{\text{train}}^{\text{plan}}$. Passenger demand P is a given input, where each passenger $p \in P$ is described with the attributes (o_p, d_p, t_p^o) that correspond to origin, destination and arrival time at the origin, respectively. The planned path r_p^{plan} of a passenger is searched by performing a shortest path algorithm on G_{plan} assuming that a passenger chooses the path with the minimum utility as shown in Eq. 10 to be the planned path.

4.2 Passenger Re-Plan Event Decision (Part II)

In part II, the disruption timetable is formulated as an event-activity network G_{dis} where the arrival and departure events form the updated ordered event list $E_{\text{train}}^{\text{dis}}$. Comparisons are made between $E_{\text{train}}^{\text{dis}}$ and the original event list $E_{\text{train}}^{\text{plan}}$ to define the set $E_{\text{train}}^{\text{cancel}}$ or $E_{\text{train}}^{\text{delay}}$, which contains all events that are cancelled or delayed during the disruption. For each passenger p whose planned path is r_p^{plan} ,

- if $r_p^{\text{plan}} \cap E_{\text{train}}^{\text{cancel}} \neq \emptyset$, then r_p^{plan} is a cancelled path either partially or completely;
- if $r_p^{\text{plan}} \cap E_{\text{train}}^{\text{cancel}} = \emptyset$ and $r_p^{\text{plan}} \cap E_{\text{train}}^{\text{delay}} \neq \emptyset$, then r_p^{plan} is a delayed path.

For the passengers whose planned paths are cancelled, they must reconsider path options. For the passengers whose planned paths are delayed only, they are also given the chance of re-planning paths in the model, while the possibility of staying with the original planned one is still kept in case no better alternative can be found. Here, the passengers whose planned paths are cancelled or delayed are called the affected passengers.

The affected passengers re-plan their paths at different locations and times, which is influenced by two factors: where they are at the moment the disruption occurs and how the information of service variations are delivered to them. The main purpose of Part II is to decide when and where an affected passenger will take the re-plan action, considering his/her location and two ways of publishing service variations, either at stations only or at both stations and trains.

Table 3 Notation

Symbol	Description
$E_{\text{train}}^{\text{plan}}$	The set of original departure and arrival events (i.e. original event list)
$E_{\text{train}}^{\text{dis}}$	The set of rescheduled departure and arrival events (i.e. updated event list)
$E_{\text{train}}^{\text{cancel}}$	The set of cancelled departure/arrival events: $E_{\text{train}}^{\text{cancel}} \subset E_{\text{train}}^{\text{plan}}$ and $E_{\text{train}}^{\text{cancel}} \cap E_{\text{train}}^{\text{dis}} = \emptyset$.
$E_{\text{train}}^{\text{delay}}$	The set of delayed departure/arrival events: $E_{\text{train}}^{\text{delay}} \cap E_{\text{train}}^{\text{plan}} = E_{\text{train}}^{\text{delay}} \cap E_{\text{train}}^{\text{dis}}$, but they differ in the times of occurrences.
$E_{\text{arr}}^{\text{dis}}$	The set of rescheduled arrival events
E_{congest}	The set of departure events that correspond to potential congested run activities
P	The set of passengers who plan to travel by train
P_{curr}	The set of passengers currently staying in the railways (either at stations or within trains)
P_{drop}	The set of passengers who drop the railways
P_{arr}	The set of passengers who arrive at the destinations by train
P_{board}	The set of passengers who want to board the same train (local variable)
P_{alight}	The set of passengers who want to alight from the same train (local variable)
P_{replan}	The set of passengers who re-plan paths upon the same arrival event (local variable)
o_p	The origin station of passenger p
d_p	The destination station of passenger p
t_p^o	The arrival time of passenger p at the origin station
r_p^{plan}	The planned path of passenger p from o_p to d_p
r_p^{dis}	The re-planned path of passenger p from the station where p re-plans to d_p
t_p^{alight}	The latest alighting time of passenger p , which is initialized with the value 0
t_p^d	The planned destination arrival time of passenger p
t_p^d	The actual destination arrival time of passenger p
B_p^{plan}	The set of departure events that correspond to the planned boarding actions of passenger p
L_p^{plan}	The set of arrival events that correspond to the planned alighting actions of passenger p
δ_p	The re-plan indication of passenger p , which indicates when and where p would re-plan the path
λ_p	The number of times of passenger p being denied for boarding a train
μ	A unique positive number that is used to indicate that a passenger would re-plan a path at the origin
st_{short}	The station where trains are short-turned or cancelled
$t_{\text{dis}}^{\text{start}}$	Start time of a disruption
$FullInfo$	The variable indicating the location of publishing train congestion information $FullInfo=Train$: publish train congestion information on trains only $FullInfo=None$: publish train congestion information nowhere
$ratio$	The specified congestion ratio that triggers updating the corresponding congestion information of a train
η	Maximum passenger accepted destination delay

4.2.1 Information of Service Variations is Published at Stations Only

Publishing service variations only at stations means that passengers can only know about the service variations at stations. Under this circumstance, a passenger would consider re-planning either at the planned origin/transfer station or at the station where his/her train is short-turned/cancelled. Figure 3 (Fig. 4) shows how to decide when and where a passenger p with a delayed (cancelled) planned path would re-plan, which is indicated by δ_p .

The basic idea of Fig. 3 is that:

- for a passenger p whose first planned boarding time at the origin is **after** the disruption start t_{dis}^{start} , p re-plans at the origin ($\delta_p = \mu$),
- for a passenger p whose first planned boarding time at the origin is **before** t_{dis}^{start} but the i th planned boarding ($i \geq 2$ here) at a transfer station happens **after** t_{dis}^{start} , p re-plans when arriving at the transfer station,
- for a passenger p whose planned boarding time at the origin is **before** t_{dis}^{start} while p has no planned transfer or the planned transfers all happen **before** t_{dis}^{start} , p will not re-plan ($\delta_p = \emptyset$).

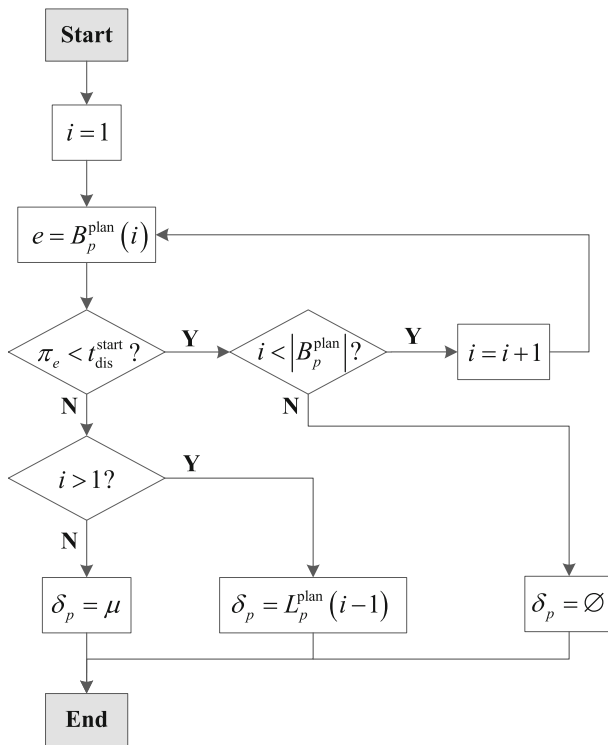


Fig. 3 Deciding the re-plan events for passengers with delayed planned paths if disruption info is published at stations only

Figure 4 shows how to decide when and where a passenger p with a cancelled planned path would re-plan. The basic idea is that:

- for a passenger p whose planned boarding time at the origin is **after** t_{dis}^{start} , p re-plans at the origin,
- for a passenger p whose planned boarding time at the origin is **before** t_{dis}^{start} but the i th planned boarding ($i \geq 2$ here) at a transfer station happens **after** t_{dis}^{start} ,
 - p re-plans when arriving at the transfer station, if the transfer station is upstream relative to the short-turn station,
 - p re-plans when being forced to get off from the train at the short-turn station, if the transfer station is downstream relative to the short-turn station,
- for a passenger p whose planned boarding times at the origin is **before** t_{dis}^{start} while p has no planned transfer or the planned transfers all happen **before** t_{dis}^{start} , p re-plans when being forced to get off from the train at the short-turn station.

4.2.2 Information of Service Variations is Published at Both Stations and Trains

Figure 5 shows how to decide when and where a passenger p with delayed/cancelled planned path would re-plan, if service variations are published at both stations and trains.

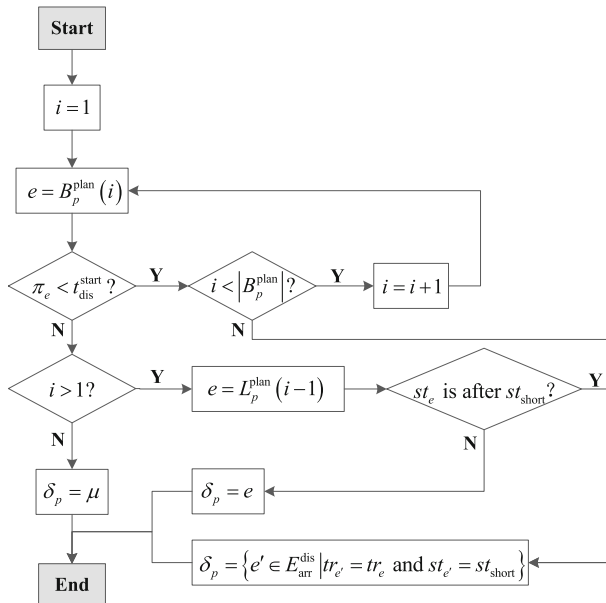


Fig. 4 Deciding the re-plan events for passengers with cancelled planned paths if disruption info is published at stations only

The basic idea of Fig. 5 is that:

- for a passenger p whose planned boarding time at the origin is **after** the disruption start t_{dis}^{start} , p re-plans at the origin,
- for a passenger p whose planned boarding time at the origin is **before** t_{dis}^{start} , the type of the latest occurring event e' of the current train tr when the disruption starts determines δ_p :
 - δ_p is set as e' , if e' is an arrival event,
 - δ_p is set as e'' of which (e', e'') is a run activity, if e' is a departure event.

4.3 Passenger Realized Path Confirmation (Part III)

In part III, the passengers' arrivals at the origins, the loading and unloading procedures and the re-plan actions are all implemented by discrete event simulation.

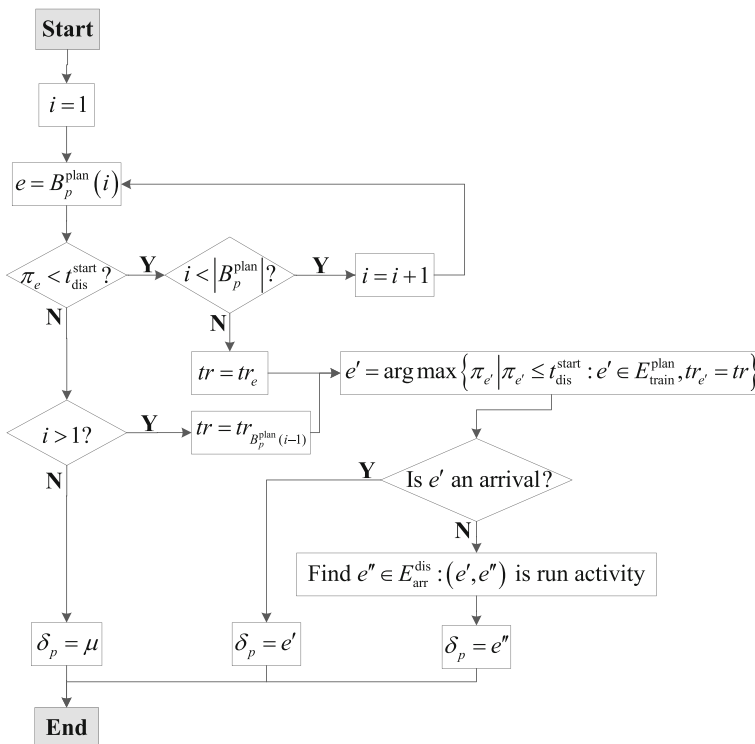


Fig. 5 Deciding the re-plan events for passengers with delayed/cancelled planned paths if disruption info is published at stations and trains

Publishing train congestion information on trains or not is considered to constrain some passengers' re-planned path choices. Note that publishing train congestion information at stations makes no sense for limiting passenger awareness of such information (i.e. all passengers can get any information published at stations), while publishing train congestion information at trains can let the passengers who are at the origins be unaware of such information. Therefore, an adaptive event-activity network G_{dis}^* is introduced, which is initialized as G_{dis} and further updated during the assignment by excluding some run activities of which the corresponding train congestions reach a specified level *ratio*. Passengers make re-planned path choices based on either G_{dis}^* or G_{dis} depending on whether they are informed with congestion information. This is explained in detail in Section 4.3.4.

In the following, the main algorithm (i.e. Discrete event passenger assignment), together with the three supporting algorithms (i.e. UpdateDep, UpdateArr and RePlan) are introduced successively.

4.3.1 Discrete Event Passenger Assignment

In Algorithm 1, different sets are initialized, and the previous system clock time is set as 0 (lines 1-3). In line 4, each event in $E_{\text{train}}^{\text{dis}}$ is iterated over to implement a passenger arrival at the origin, and the loading or unloading procedure. $E_{\text{train}}^{\text{dis}}$ is a given input, of which the contained events are previously sorted in time-ascending order and then in alphabetical order regarding the event type (arrival or departure), to ensure that the assignment proceeds with time, and an arrival event occurs before a departure event if their time instants are the same. In lines 5-6, the first element from $E_{\text{train}}^{\text{dis}}$ is chosen as the current event e to be executed, and the current system clock time is set as the occurrence time of e . In the loop starting from line 7, each arrival of a passenger at the origin between the previous and the current system clock time is simulated. In line 8, the origin arrival passenger p is included to the set P_{curr} that contains all passengers who are currently staying in the railways. If p needs to re-plan at the origin, the function RePlan will be called to realize the re-plan action (lines 9-10). In line 11, p is excluded from P to avoid being included in P_{curr} again.

If the current event e is a departure (line 12), then the passengers who want to board train tr_e are defined by P_{board} (line 13). If P_{board} is not empty (line 14), it is excluded from the set P_{curr} (line 15) and the loadings of passengers in P_{board} are implemented by calling the function UpdateDep (line 16). In line 17, the updated P_{board} is included to P_{curr} again. The reason of having lines 15 and 17 is that when executing UpdateDep, some passengers could be removed from P_{board} to P_{drop} , because they drop the railways due to denied boardings and no preferred alternatives can be found. Function UpdateDep also outputs E_{congest} and G_{dis}^* , which are the set of departure events that correspond to potential congested run activities, and an adaptive event-activity network that excludes the potential congested run activities from G_{dis} .

Algorithm 1 Discrete event passenger assignment.

Input: $E_{\text{train}}^{\text{dis}}, P, t_{\text{dis}}^{\text{start}}, G_{\text{dis}}, \text{ratio}, \text{FullInfo}, \eta$
Output: $P_{\text{arr}}, P_{\text{drop}}$

```

1   $G_{\text{dis}}^* = G_{\text{dis}};$ 
2  Let  $P_{\text{curr}}, P_{\text{arr}}, P_{\text{drop}}$  and  $E_{\text{congest}}$  all be empty ;
3   $\text{clock}_0 \leftarrow 0;$ 
4  while  $E_{\text{train}}^{\text{dis}} \neq \emptyset$  do
5       $e \leftarrow E_{\text{train}}^{\text{dis}}(1);$ 
6       $\text{clock}_1 \leftarrow \pi_e;$ 
7      foreach  $p \in \{p' \in P : \text{clock}_0 < t_{p'}^0 \leq \text{clock}_1\}$  do
8           $P_{\text{curr}} \leftarrow P_{\text{curr}} \cup \{p\};$ 
9          if  $\delta_p = \mu$  then
10              $(P_{\text{curr}}, P_{\text{drop}}) = \text{RePlan}(p, e, t_{\text{dis}}^{\text{start}}, P_{\text{curr}}, P_{\text{drop}}, G_{\text{dis}}, G_{\text{dis}}^*, Z_1, \text{FullInfo}, \eta);$ 
11              $P \leftarrow P \setminus \{p\};$ 
12  if  $e$  is a departure event then
13      Find the set  $P_{\text{board}} := \{p \in P_{\text{curr}} \mid B_p(1) = e\};$ 
14      if  $P_{\text{board}} \neq \emptyset$  then
15           $P_{\text{curr}} \leftarrow P_{\text{curr}} \setminus P_{\text{board}};$ 
16           $(P_{\text{board}}, P_{\text{drop}}, E_{\text{congest}}, G_{\text{dis}}^*) =$ 
17              $\text{UpdateDep}(e, t_{\text{dis}}^{\text{start}}, P_{\text{board}}, P_{\text{drop}}, G_{\text{dis}}, G_{\text{dis}}^*, \text{ratio}, \text{FullInfo}, E_{\text{congest}}, \eta);$ 
18              $P_{\text{curr}} \leftarrow P_{\text{curr}} \cup P_{\text{board}};$ 
19  else if  $e$  is an arrival event then
20      Find the set  $P_{\text{alight}} := \{p \in P_{\text{curr}} \mid L_p(1) = e\}$  and the set  $P_{\text{replan}} := \{p \in P_{\text{curr}} \mid \delta_p = e\};$ 
21      if  $P_{\text{alight}} \cup P_{\text{replan}} \neq \emptyset$  then
22           $P_{\text{curr}} \leftarrow P_{\text{curr}} \setminus (P_{\text{alight}} \cup P_{\text{replan}});$ 
23           $(P_{\text{alight}}, P_{\text{replan}}, P_{\text{drop}}, P_{\text{arr}})$ 
24              $= \text{UpdateArr}(e, t_{\text{dis}}^{\text{start}}, P_{\text{alight}}, P_{\text{replan}}, P_{\text{drop}}, P_{\text{arr}}, G_{\text{dis}}, G_{\text{dis}}^*, \text{FullInfo}, \eta);$ 
25              $P_{\text{curr}} \leftarrow P_{\text{curr}} \cup (P_{\text{alight}} \cup P_{\text{replan}});$ 
26      if  $\text{FullInfo} = \text{Train}$  then
27          foreach  $p \in P_{\text{curr}} \setminus (P_{\text{alight}} \cup P_{\text{replan}})$  do
28              Let  $r$  be  $r_p^{\text{dis}}$  if  $r_p^{\text{dis}}$  is nonempty. Otherwise, let  $r$  be  $r_p^{\text{plan}};$ 
29              if  $e \in r$  and  $B_p(1) \in E_{\text{congest}}$  then
30                   $(P_{\text{curr}}, P_{\text{drop}}) = \text{RePlan}(p, e, t_{\text{dis}}^{\text{start}}, P_{\text{curr}}, P_{\text{drop}}, G_{\text{dis}}, G_{\text{dis}}^*, Z_4, \text{FullInfo}, \eta);$ 
31   $\text{clock}_0 \leftarrow \text{clock}_1;$ 
32   $E_{\text{train}}^{\text{dis}} \leftarrow E_{\text{train}}^{\text{dis}} \setminus \{e\};$ 
33  return  $P_{\text{arr}}, P_{\text{drop}}$ 

```

If the current event e is an arrival (line 18), the passengers who want to alight from the arriving train are defined by P_{alight} and the passengers who will re-plan paths when e occurs are defined by P_{replan} (line 19). If at least one of the two sets is not empty (line 20), the union $P_{\text{alight}} \cup P_{\text{replan}}$ is excluded from P_{curr} first (line 21) and then the function UpdateArr is called to implement the unloadings of passengers in P_{alight} and the re-plan actions of passengers in P_{replan} (line 22). In line 23, the

updated $P_{\text{alight}} \cup P_{\text{replan}}$ is included to P_{curr} again. The reason of having lines 21 and 23 is that when executing UpdateArr, some passengers could be removed from P_{alight} to P_{arr} because they reach the destinations, and some passengers could be removed from P_{replan} to P_{drop} because they cannot find preferred re-planned paths and thus drop the railways. If train congestion information is published on trains (line 24), for each passenger who is dwelling at train tr_e and the next boarding train is highly congested as notified (lines 25-27), the passenger is given the chance of re-planning (line 28).

After finishing executing the current event, the previous system clock time is set to the current system clock time, and the current event is removed from the event list to be executed (lines 29-30).

4.3.2 Passenger Loading

Algorithm 2 UpdateDep.

Input: $e, t_{\text{dis}}^{\text{start}}, P_{\text{board}}, P_{\text{drop}}, G_{\text{dis}}, G_{\text{dis}}^*, ratio, FullInfo, E_{\text{congest}}, \eta$
Output: $P_{\text{board}}, P_{\text{drop}}, E_{\text{congest}}, G_{\text{dis}}^*$

```

1  if  $cap_{tr_e} \geq |P_{\text{board}}|$  then
2       $cap_{tr_e} \leftarrow cap_{tr_e} - |P_{\text{board}}|$ ;
3      foreach  $p \in P_{\text{board}}$  do
4           $B_p \leftarrow B_p \setminus \{e\}$ ;
5  else if  $cap_{tr_e} < |P_{\text{board}}|$  and  $cap_{tr_e} > 0$  then
6       $cap_{tr_e} \leftarrow 0$ ;
7      Sort  $P_{\text{board}}$  in ascending order according to  $\max \{t_p^o, t_p^{\text{alight}}\}$ ;
8      foreach  $p \in P_{\text{board}}(1 : cap_{tr_e})$  do
9           $B_p \leftarrow B_p \setminus \{e\}$ ;
10     foreach  $p \in P_{\text{board}}(cap_{tr_e} + 1 : \text{end})$  do
11          $\lambda_p \leftarrow \lambda_p + 1$ ;
12          $(P_{\text{board}}, P_{\text{drop}}) = \text{RePlan}(p, e, t_{\text{dis}}^{\text{start}}, P_{\text{board}}, P_{\text{drop}}, G_{\text{dis}}, G_{\text{dis}}^*, Z_2, FullInfo, \eta)$ ;
13  else if  $cap_{tr_e} = 0$  then
14      foreach  $p \in P_{\text{board}}$  do
15           $\lambda_p \leftarrow \lambda_p + 1$ ;
16           $(P_{\text{board}}, P_{\text{drop}}) = \text{RePlan}(p, e, t_{\text{dis}}^{\text{start}}, P_{\text{board}}, P_{\text{drop}}, G_{\text{dis}}, G_{\text{dis}}^*, Z_2, FullInfo, \eta)$ ;
17  if  $Fulltrain = \text{Train}$  then
18      if  $(1 - cap_{tr_e}/cap_{tr_e}^{\text{max}}) \geq ratio$  then
19          Find the next run activity of train  $tr_e$ :  $a = (e', e'')$ ;
20           $E_{\text{congest}} \leftarrow E_{\text{congest}} \cup \{e'\}$ ;
21           $G_{\text{dis}}^* \leftarrow G_{\text{dis}}^* \setminus \{a\}$ ;
22  return  $P_{\text{board}}, P_{\text{drop}}, G_{\text{dis}}^*, E_{\text{congest}}$ 

```

In algorithm 2, if the available capacity cap_{tr_e} of a train tr_e is sufficient to cover all passengers P_{board} who want to board the train, cap_{tr_e} is updated accordingly (lines 1-2). Then, for each $p \in P_{\text{board}}$, the current event e is excluded from the set B_p that contains all departure events corresponding to the boarding actions of p (lines 3-4).

If the available capacity of a train can only cover part of the passengers in P_{board} (line 5), cap_{tr_e} is updated accordingly (line 6) and then the passengers in P_{board} are sorted in ascending order according to their arriving times at the current stations (line 7). Here, t_p^{alight} refers to the latest alighting time of passenger p , of which the value is initialized with 0 when p arrives at the origin and further be updated when p alights from a train at another station. Line 7 ensures the loading rule of first-come-first-served. The first cap_{tr_e} passengers in P_{board} can board the train (lines 8-9), while the remainders are denied for boarding and RePlan is called for re-planning (line 10-12). If the available capacity of a train is zero, none of the passengers in P_{board} can board the train, but only re-plan paths (lines 13-16).

Furthermore, if train congestion information is published and the congestion ratio of the train tr_e , $(1 - cap_{tr_e}/cap_{tr_e}^{\text{max}})$, has reached the specified congestion level *ratio*, then the departure event e' that corresponds to the next run activity a of tr_e is added to E_{congest} (lines 19-20), while a is excluded from the adaptive event activity network G_{dis}^* (line 21). Here, $cap_{tr_e}^{\text{max}}$ represents the maximum number of passengers that train tr_e can hold.

4.3.3 Passenger Unloading

Algorithm 3 UpdateArr.

Input: $e, t_{\text{dis}}^{\text{start}}, P_{\text{alight}}, P_{\text{replan}}, P_{\text{drop}}, P_{\text{arr}}, G_{\text{dis}}, G_{\text{dis}}^*, \text{FullInfo}, \eta$
Output: $P_{\text{alight}}, P_{\text{replan}}, P_{\text{drop}}, P_{\text{arr}}$

```

1  foreach  $p \in P_{\text{alight}}$  do
2     $cap_{tr_e} \leftarrow cap_{tr_e} + 1$ ;
3     $L_p \leftarrow L_p \setminus \{e\}$ ;
4    if  $L_p = \emptyset$  then
5       $t_p^d = \pi_e$ ;
6       $P_{\text{alight}} \leftarrow P_{\text{alight}} \setminus \{p\}$ ;
7       $P_{\text{arr}} \leftarrow P_{\text{arr}} \cup \{p\}$ ;
8    else
9       $t_p^{\text{alight}} = \pi_e$ ;
10 foreach  $p \in P_{\text{replan}}$  do
11    $(P_{\text{replan}}, P_{\text{drop}}) = \text{RePlan}(p, e, t_{\text{dis}}^{\text{start}}, P_{\text{replan}}, P_{\text{alight}}, P_{\text{drop}}, G_{\text{dis}}, G_{\text{dis}}^*, Z_3, \text{FullInfo}, \eta)$ ;
12 return  $P_{\text{alight}}, P_{\text{replan}}, P_{\text{drop}}, P_{\text{arr}}$ 

```

In algorithm 3, for each passenger p who wants to alight from the train tr_e , the available capacity of the train is updated accordingly (lines 1-2), and event e is excluded from the set L_p that contains all arrival events corresponding to the alighting actions of p (line 3). After that, an empty L_p means that passenger p has reached the destination (line 4), thus the actual destination arrival time t_p^d is updated and p is removed from P_{alight} to P_{arr} (lines 5-7). If L_p is not empty, the latest alighting time t_p^{alight} is updated (lines 8-9). For each passenger who will re-plan path when e occurs, RePlan is called (lines 10-11).

4.3.4 Passenger Re-Planning

Algorithm 4 RePlan.

Input: $p, e, t_{\text{dis}}^{\text{start}}, P_{\text{board}}$ (or P_{replan} and P_{alight} , or P_{curr}), $P_{\text{drop}}, G_{\text{dis}}, G_{\text{dis}}^*, Z_j, \text{FullInfo}, \eta$
Output: P_{board} (or P_{replan}), P_{drop}

- 1 Determine the source node u and sink node v of passenger p according to d_p, Z_j, t_p^o and st_e ;
- 2 Determine whether G is set as G_{dis} or G_{dis}^* according to Z_j and FullInfo ;
- 3 Search the optimal path r from u to v on G ;
- 4 Let t_r^d be the destination arrival time of path r ;
- 5 **if** $(t_r^d - \hat{t}_p^d) \leq \eta$ **then**
- 6 $r_p^{\text{dis}} \leftarrow r$;
- 7 Update B_p and L_p according to r_p^{dis} ;
- 8 **if** $Z_j \in \{Z_3, Z_4\}$ **and** $p \notin P_{\text{alight}}$ **then**
- 9 **if** $L_p(1) = e$ **then**
- 10 $\text{cap}_{tr_e} \leftarrow \text{cap}_{tr_e} + 1$;
- 11 $L_p \leftarrow L_p \setminus \{e\}$;
- 12 $t_p^{\text{alight}} = \pi_e$;
- 13 **else**
- 14 $P_{\text{drop}} \leftarrow P_{\text{drop}} \cup \{p\}$;
- 15 P_{board} (or P_{replan} , or P_{curr}) $\leftarrow P_{\text{board}} \setminus \{p\}$ (or $P_{\text{replan}} \setminus \{p\}$, or $P_{\text{curr}} \setminus \{p\}$);
- 16 **return** P_{board} (or P_{replan} , or P_{curr}), P_{drop}

In line 1, the source and sink nodes (i.e. u and v) are determined for searching the re-planned path of passenger p (line 1). The sink node is always the exit event corresponding to the destination d_p , while the source node u is different under different re-plan situations.

- If $Z_j = Z_1$, u is set as the duplicate departure event that is closest to the passenger's arrival time at the origin: t_p^o .
- If $Z_j = Z_2$, u is set as the duplicate departure event that is closest to the current departure event e at the station st_e .
- If $Z_j = Z_3 \cup Z_4$, u is set as the current arrival event e .

All re-plan situations are listed in Table 4.

In line 2, the event-activity network G that is used to search the re-planned path of passenger p , is determined according to the values of FullInfo and Z_j .

- If $\text{FullInfo} = \text{Train}$, then train congestion information is published at trains only. Thus, G is set as G_{dis}^* ,

Table 4 Re-plan situations

Situation	Time and location
Z_1 : re-plan before travelling due to service variations	When arriving at the origin station
Z_2 : re-plan during travelling due to denied boarding	When planning to board at any possible station
Z_3 : re-plan during travelling due to service variations	When arriving at the specified station
Z_4 : re-plan during travelling due to train congestion	When arriving at a station

- if $Z_j = Z_2$ and $st_e \neq o_p$, which means passenger p re-plan paths due to denied boarding at the station that is not his/her origin, or
- if $Z_j \in \{Z_3, Z_4\}$.

Passenger who satisfies either of the above conditions must have taken a train where he/she is notified with train congestion information.

- If $FullInfo=none$, then train congestion information is published nowhere. Thus, G is always set as G_{dis} whatever Z_j is.

In line 3, the optimal path r is searched through G , of which the destination arrival time is t_r^d (line 4). Thus, the resulting delay of r is $(t_r^d - \hat{t}_p^d)$ where \hat{t}_p^d is the planned destination arrival time of p .

If the delay is no longer than passenger's maximum accepted delay η (line 5), r is chosen as the re-planned path (line 6), and the sets of events corresponding to the boardings and alightings are updated accordingly (line 7). Additionally for the re-plan situation of Z_3 or Z_4 , a passenger who does not plan to alight from the train tr_e might now want to get off due to the re-planned path (lines 8-9). In such a case, the available train capacity, the set of alighting events, and the latest alighting time are all updated accordingly (lines 10-12).

If the delay of r is longer than η (line 13), the passenger will drop the railway. Thus, the set P_{drop} and the set P_{board} (or P_{replan}) are all updated accordingly (lines 14-15).

5 Time Complexity

Algorithm 1, the main algorithm working for passenger assignment, is based on three sub-algorithms (i.e. Algorithms 2,3 and 4), while Algorithms 2 and 3 also need to call Algorithm 4. Figure 6 shows the relations between the algorithms.

In RePlan, lines 1, 3 and 7 require non-constant time. Line 1 takes $O(|E_{ddep}|)$ time, where E_{ddep} represents the set of duplicate departure events. Line 3 takes $O(|A| + |E|)$ time, where A and E refer to the activities and events contained in the formulated event-activity network, respectively. Line 7 takes $O(|r_p^{dis}|)$ time, where

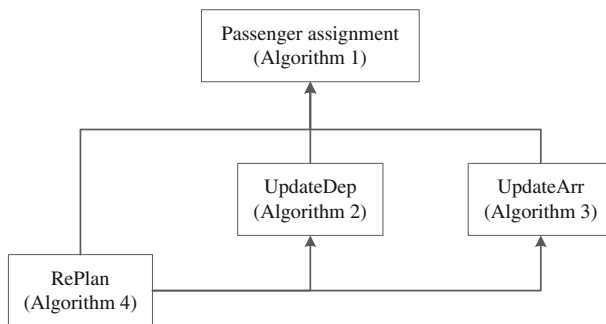


Fig. 6 The relations between algorithms

r_p^{dis} refers to the re-planned path of passenger p , which contains all events that p will pass through. Thus, the time complexity of Algorithm 4 is

$$\begin{aligned} T_{Alg}^4 &= O(|E_{\text{ddep}}|) + O(|A| + |E|) + O\left(|r_p^{\text{dis}}|\right), \\ &\leq O(|E|) + O(|A| + |E|) + O(|E|), \\ &= O(|A| + |E|). \end{aligned}$$

In UpdateArr, the **for** loop from line 1 to line 9 takes $O(|P_{\text{alight}}|)$ time, while the **for** loop from line 10 to line 11 takes $O(|P_{\text{replan}}| \cdot (|A| + |E|))$ time. Thus, the time complexity of Algorithm 3 is

$$T_{Alg}^3 = O(|P_{\text{alight}}|) + O(|P_{\text{replan}}| \cdot (|A| + |E|)).$$

In UpdateDep, line 7 takes $O(|P_{\text{board}}| \log |P_{\text{board}}|)$ time by using heapsort. Thus, lines 1-16 take $O(|P_{\text{board}}| \log |P_{\text{board}}|) + O(|P_{\text{board}}| \cdot (|A| + |E|))$ time. Because $\log |P_{\text{board}}| \ll |A| + |E|$ and lines 17-21 take constant time, the time complexity of Algorithm 2 is

$$T_{Alg}^2 = O(|P_{\text{board}}| \cdot (|A| + |E|)).$$

As for Algorithm 1, the **while** loop makes one iteration per event of $E_{\text{train}}^{\text{dis}}$. Let N be the size of $E_{\text{train}}^{\text{dis}}$, thus in all N **while** iterations:

- the **for** loop from line 7 to line 11 takes at most $\sum_{i=1}^N |P_{\text{ori}}^i| \cdot (|A| + |E|)$ operations, where P_{ori}^i represents the set of passengers who arrive at the origins in the i th iteration. This is equal to $O(|P| \cdot (|A| + |E|))$ time, where P represents the total passenger demand.
- due to the calls of UpdateDep, lines 12 - 17 take at most $\sum_{i=1}^N |P_{\text{board}}^i| \cdot (|A| + |E|)$ operations, where P_{board}^i represents the set of passengers who wish to board the train at the i th iteration. By defining m as the maximum number of boardings/alightings that a passenger may encounter, there must be $\sum_{i=1}^N |P_{\text{board}}^i| = m|P|$. Because the maximum number of boardings/alightings corresponding to a passenger must be finite and relatively small, the time complexity is $O(|P| \cdot (|A| + |E|))$.
- due to the calls of UpdateArr, lines 18-23 take $\sum_{i=1}^N |P_{\text{alight}}^i| + |P_{\text{replan}}^i| \cdot (|A| + |E|)$ operations, where P_{alight}^i (P_{replan}^i) represents the set of passengers who will alight from the train (re-plan paths) at the i th iteration. This is not larger than $m|P| + |P| \cdot (|A| + |E|)$. Thus, lines 18-23 totally take $O(|P| \cdot (|A| + |E|))$ time.
- lines 5-6 and lines 29-30 totally take $O(N)$ time, where N is smaller than $|E|$.

If train congestion information is not published, the time complexity of Algorithm 1 thus is

$$\begin{aligned} T_{Alg}^1 &= O(|P| \cdot (|A| + |E|)) + O(|P| \cdot (|A| + |E|)) + O(|P| \cdot (|A| + |E|)) + O(N), \\ &= O(|P| \cdot (|A| + |E|)). \end{aligned}$$

If train congestion information is published, in each **while** iteration of Algorithm 1, the **for** loop from line 25 to line 28, at most take $|P|$ operations in case

Table 5 Notations used in proving the time complexity of Algorithm 1

Symbol	Description
P_{ori}^i	The set of passengers who arrive at the origins at the i th while iteration of Algorithm 1
P_{board}^i	The set of passengers who wish to board the train at the i th while iteration of Algorithm 1
P_{align}^i	The set of passengers who will alight from the train at the i th while iteration of Algorithm 1
P_{replan}^i	The set of passengers who re-plan paths upon arrival event due to service variations at the i th while iteration of Algorithm 1
m	The maximum number of boardings/alightings a passenger could encounter

$P_{curr} = P$ and $(P_{align} \cup P_{replan}) = \emptyset$. In such a case, T_{Alg}^1 becomes $O(N \cdot |P|) + O(|P| \cdot (|A| + |E|))$. As $N < |A| + |E|$, T_{Alg}^1 is still $O(|P| \cdot (|A| + |E|))$, although train congestion information is published.

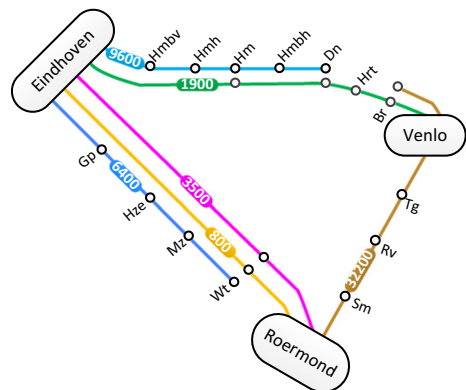
The notation used is given in Table 5.

In summary, the time complexity of the proposed passenger assignment model is relevant to the size of the given passenger demand and the scale of the considered network. To reduce computational burden, one way is to group the passengers who share the same travel characteristics (e.g. the origins, the destinations, the arrival times at the origin, etc.). However, this is at the expense of assignment accuracy, since two passengers who have exactly the same travel characteristics could still be distributed to different trains if vehicle capacities are in short supply.

6 Case Study

6.1 Description

The model is applied to a subnetwork of the Dutch railways, where 17 stations are located and six train lines are operated (see Fig. 7). The disruption scenario is defined as a complete track blockage between stations Hze and Mz, which occurs from 7:57

Fig. 7 The considered network


to 9:00. Here, only the disruption timetable of the corridor where the disruption happens is shown (see Fig. 8), as the timetables of the other two corridors remain as planned. The number of nodes (events) and arcs (activities) in the network formulation are 2085 and 3539, respectively. In Fig. 8, the solid lines represent the services scheduled in the disruption timetable, while the dashed (dotted) lines represent the original scheduled services that are cancelled (delayed) in the disruption timetable.

Passenger demand is generated for the period from 7:00 to 10:00, which contains the time frame before the disruption starts, during the disruption and after the disruption ends, since passengers who start travelling during these hours could be influenced by the disruption. The total number of passengers who travel in the considered network during the considered period is 7515.

To consider information interventions, different schemes of information provision are set, which are listed in Table 6. Each row of Table 6 indicates the locations where service variations and train congestion information are published (i.e. *ServiceInfo* and *FullInfo*), the specified train congestion level (i.e. *ratio*) that triggers updating the congestion information, and the maximum accepted destination delay of each passenger (i.e. η). For example the current congestion ratio of a train is 0.85; if *ratio* is set as 0.85, then the operators update the congestion information by telling passengers that the next run of this train would be highly congested. However, if *ratio* is set as 0.9, no such information will be given to passengers, because the current congestion ratio of the train, 0.85, does not reach the level of 0.9. A train of which the congestion ratio currently reaches *ratio*, is thought to be potentially unable to satisfy all boarding demands for its next run. Thus, notifying some passengers with the congestion information can avoid them boarding the next run of the train, while some passengers who are not notified with information may still keep their choices. As for η , two kinds of values are set here, which are $(\hat{t}_p^d - t_p^o)$ referring to the planned travel time

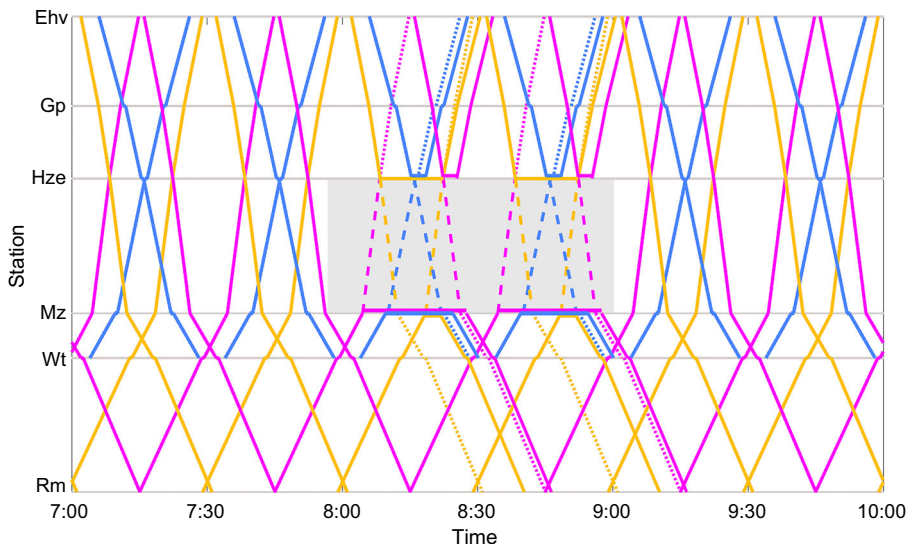


Fig. 8 Disruption timetable

Table 6 Case study settings

<i>ServiceInfo</i>	<i>FullInfo</i>	<i>ratio</i>	η [min]
Station	Train	0.8,0.9,or 1	$(\hat{t}_p^d - t_p^o)$ or 63
Station	None	–	$(\hat{t}_p^d - t_p^o)$ or 63
Station & Train	Train	0.8,0.9,or 1	$(\hat{t}_p^d - t_p^o)$ or 63
Station & Train	None	–	$(\hat{t}_p^d - t_p^o)$ or 63
None	None	–	$(\hat{t}_p^d - t_p^o)$ or 63

of passenger p , and 63 min which is the defined disruption duration. The value of η may affect the number of passengers who drop the railways, which further affects the congestion of trains. Also baseline scenarios are created, in which neither service variations nor train congestion is provided. In these scenarios, passengers can only know the service variation when they really experience it themselves. For instance, a passenger knows he/she cannot board a train as planned when the train does not show up at the station due to delay or cancellation.

6.2 Results

By applying the model on the defined disruption scenarios with the settings of Table 6, 18 results are obtained, which are shown in Table 7. In each result, three indicators are used, which are the number of dropped passengers, the number of denied boardings, and the travel time deviation. The computation time for scenario 9 is the least, which is 15 seconds, and the computation time for scenario 16 is the most, which is 26 seconds. The computation times for other scenarios are between these two.

The number of dropped passengers is calculated as $|P_{\text{drop}}|$, and the number of denied boardings is calculated as $\sum_{p \in P} \lambda_p$ where λ_p represents the number of times a passenger being denied for boarding and $P = P_{\text{arr}} \cup P_{\text{drop}}$. The travel time deviation is calculated as

$$\begin{aligned}
 & - \sum_{p \in P_{\text{arr}}} (\hat{t}_p^d - t_p^d) + \sum_{p \in P_{\text{drop}}} (\hat{t}_p^d - t_p^o), \text{ when } \eta \text{ is set as } \hat{t}_p^d - t_p^o, \text{ or} \\
 & - \sum_{p \in P_{\text{arr}}} (\hat{t}_p^d - t_p^d) + 63 |P_{\text{drop}}|, \text{ when } \eta \text{ is set as } 63,
 \end{aligned}$$

where $\hat{t}_p^d(t_p^d)$ represents the actual(planned) destination arrival time of passenger p , and t_p^o represents the actual origin arrival time of passenger p . Note that t_p^d could be smaller or larger than \hat{t}_p^d , which means the total travel time deviation consists of both negative and positive individual travel time deviations.

6.2.1 Influence of Maximum Accepted Delay: η

It is found from Table 7 that the total travel time deviations in all cases are positive, which means that the actual travel times increase compared to the planned travel times. When the maximum accepted destination delay (i.e. η) is set as an individual's

Table 7 Results of disruption scenarios between stations Hze and Mz

Scenario	η [min]	<i>ServiceInfo</i>	<i>FullInfo</i>	<i>ratio</i>	# drop passengers	# denied boardings	Travel time deviation [min]
1	$\hat{t}_p^d - t_p^o$	Station	Train	0.8	551	0	25243
2	$\hat{t}_p^d - t_p^o$	Station	Train	0.9	551	0	25243
3	$\hat{t}_p^d - t_p^o$	Station	Train	1	551	0	25243
4	$\hat{t}_p^d - t_p^o$	Station	None	–	551	0	25243
5	$\hat{t}_p^d - t_p^o$	Station & Train	Train	0.8	551	0	25187
6	$\hat{t}_p^d - t_p^o$	Station & Train	Train	0.9	551	0	25187
7	$\hat{t}_p^d - t_p^o$	Station & Train	Train	1	551	0	25187
8	$\hat{t}_p^d - t_p^o$	Station & Train	None	–	551	0	25187
9	$\hat{t}_p^d - t_p^o$	None	None	–	565	0	28399
10	63	Station	Train	0.8	118	178	39758
11	63	Station	Train	0.9	113	193	40687
12	63	Station	Train	1	113	193	40693
13	63	Station	None	–	113	193	40693
14	63	Station & Train	Train	0.8	116	178	39707
15	63	Station & Train	Train	0.9	110	193	40590
16	63	Station & Train	Train	1	110	193	40595
17	63	Station & Train	None	–	110	193	40595
18	63	None	None	–	88	358	47234

planned travel time (i.e. $\hat{t}_p^d - t_p^o$), the travel time increases are the smallest, while the numbers of dropped passengers (i.e. 551 or 565) are the largest (scenarios 1-9). When η is set as the disruption length (scenarios 10-18), the travel time increases grow, while the numbers of dropped passengers reduce. These indicate that

- the provided disruption timetable leads to 565 passengers to whom the increased travel time in the railways is at least equal to the planned travel time, and
- the limited vehicle capacities lead to more travel time increase when more passengers remain in the railways to reach the destinations.

If operators aim for a low travel time increase while satisfying passenger demand as much as possible, one way is to design a disruption timetable that provides faster services and ensures less denied boardings by adjusting the schedules to distribute passengers wisely. However, this is rather challenging, since many factors (e.g. passenger behaviour, vehicle capacities, infrastructure restrictions, etc.) need to be considered in the rescheduling. Thus, another way is proposed, information intervention, which is easier to be implemented in practice. Here, information intervention means that operators provide passengers at different locations with different information about service variations or train congestion.

6.2.2 Influence of Information Intervention

When η is set as $\hat{t}_p^d - t_p^o$ or 63 min, compared to not updating passengers with any information (scenario 9 or 18), providing information for them (scenarios 1-8 or 10-17) helps to reduce the number of dropped passengers, the number of denied boardings, and/or the travel time increase. This indicates that it is helpful to update passengers with certain information during disruptions.

When η is set as $\hat{t}_p^d - t_p^o$, providing service variations at both stations and trains (scenarios 5-8) always lead to less travel time increases compared to the cases in which service variations are announced at stations only (scenarios 1-4). However, publishing train congestion information does not make any sense, since no one has been denied for boarding even though train congestion information is not published. This indicates that

- if vehicle capacities are not in short supply, publishing service variations at both stations and trains is able to reduce more travel time increase, compared to publishing service variations at stations only.

This is because by additionally receiving service variations on trains, the on-board passengers at the moment the disruption occurs can re-plan paths just at the next stop of the train rather than several stops later where they get off from the train, and such earlier re-plans help to find better alternatives which result in less travel time increases.

When η is set as 63 min, more passengers remained in the railways and thus some passengers were denied for boarding due to insufficient vehicle capacities. Under this circumstance, additionally publishing train congestion information on trains helps to reduce travel time increase, if *ratio* is set to an appropriate value. For example when service variations are published at stations only, compared to not publishing train congestion (scenario 13), publishing such information on trains leads to better performance when *ratio* is set as 0.8 or 0.9 (scenario 10 or 11), or does not change performance when *ratio* is set as 1 (scenario 12). These phenomena are also found in the cases where service variations are published at both stations and trains (scenarios 14-17). These indicate that

- if vehicle capacities are in short supply, the performance of publishing train congestion on reducing travel time increase is influenced by the value of *ratio*.

For example if *ratio* is set as 0.8 and the current congestion ratio of train tr_1 is 0.85, then the information that tr_1 is highly congested is published to on-board passengers, which prevent them from boarding the next run of tr_1 , while the passengers who wait at their origins to board the next run of tr_1 still keep their choices. In this way, passengers who demand for boarding the next run of tr_1 are distributed, since tr_1 is thought to be highly congested now and may be unable to satisfy all demands later. If there must be some passengers being denied for boarding a train, avoiding them to choose the train may help them find better alternatives compared to the ones they can find after being denied. It is also possible that all subsequent demands are satisfied if these demands are small, or lots of passengers get off before the next run, thus avoiding some passengers boarding the train might not be helpful since the avoided

passengers may not be able to find better alternatives. Clearly, the setting of *ratio* is important, which decides whether publishing train congestion is good or not. One way to ensure the accuracy of *ratio* is to assign each train with a customized *ratio* that varies with times and locations according to the estimated boarding demand and alighting amount. However these are also hard to be estimated accurately, as during the period from the departure of the current run to the departure of the next run, there could be extra demand or alightings emerging. In this paper, a general value of *ratio* is set to all trains, while it would be interesting to enable a dynamic customized *ratio* to each train in future research.

6.2.3 Impacts on Passengers Who Start Travelling Before/During the Disruption

In the model, the passengers who start travelling before the disruption, but are still within the railways at the moment the disruption occurs, are also considered explicitly. According to the case study results, these passengers are affected a lot by the disruption, thus overlooking them in the assignment model is unreasonable. In Figs. 9 and 10, the individual delays of scenario 1 and 10, are shown respectively, which are distinguished by passengers' travelling start time. As few passengers who start travelling after the disruption ends (i.e. 9:00) are delayed (9 individuals in scenario 1 or scenario 10), only the individual delays of passengers who start travelling before or during the disruption are shown. It is found that under the same setting of η (i.e. the maximum acceptable destination delay), individual delays are not significantly different. Thus, scenario 1 is chosen as the representative of setting η to individual planned travel time, and scenario 10 is chosen as the representative of setting η to the defined disruption length. Individual delay is only calculated for the passenger who

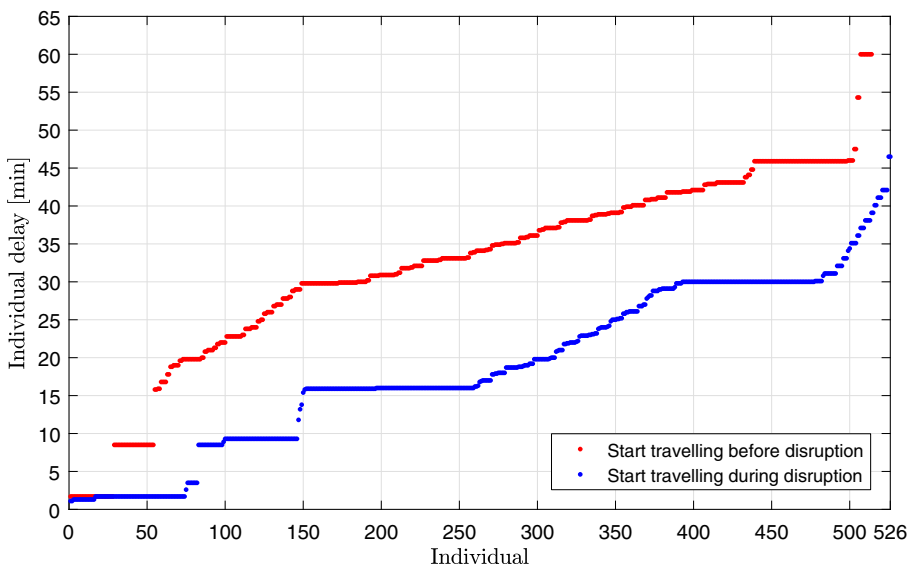


Fig. 9 Individual delays of scenario 1 (η =individual planned travel time)

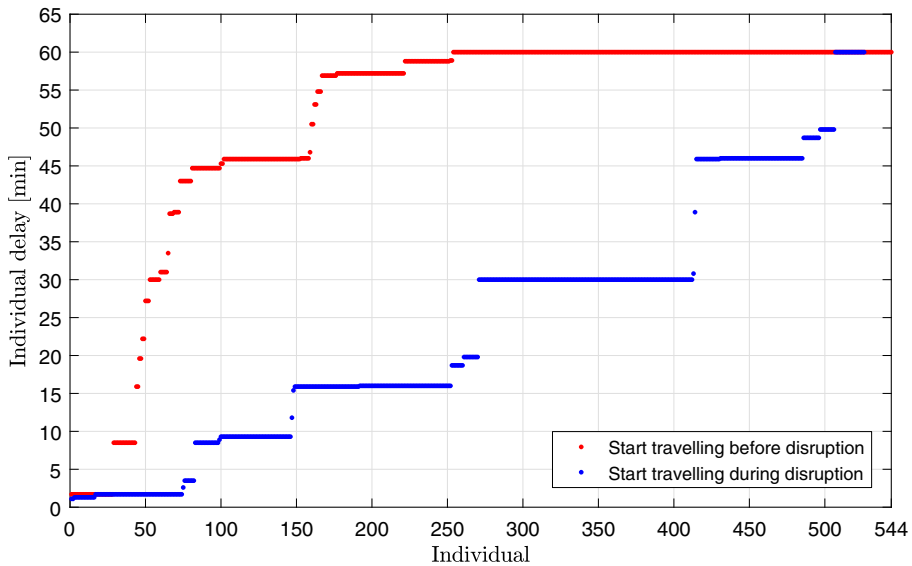


Fig. 10 Individual delays of scenario 10 (η =disruption length)

actually arrived at the destination later than planned. Thus, individual delay is calculated as $t_p^d - \hat{t}_p^d$ for $p \in P_{\text{arr}}$, if $t_p^d > \hat{t}_p^d$. For $p \in P_{\text{drop}}$, individual delay is computed as the corresponding value of η . Note that in Figs. 9 and 10, individual delays are shown in ascending order, and an individual numbered in Fig. 9 does not correspond to the individual numbered the same in Fig. 10.

Figures 9 and 10 both show that passengers who start travelling before the disruption are delayed more seriously. Most individual delays are below 50 minutes in Fig. 9, while in Fig. 10, there are a lot of individual delays reaching 60 minutes. This indicates that most passengers' planned travel times are below 50 minutes and it is hard for them to find the alternatives of less than 50 minutes delay under the current disruption timetable, while the congestion issue increases passenger delays further. To reduce passenger delays, one way is to improve the disruption timetable by providing passengers with better alternatives (i.e. less resulting delays), which could be done by incorporating passenger responses into timetable rescheduling. Additionally, it is found that there are 21 passengers in scenario 10 being delayed, not because of service variations (i.e. their planned paths are not cancelled/delayed) but due to denied boardings only. Under these circumstances, increasing vehicle capacities or providing alternatives outside the railways (e.g. shuttle buses), would be helpful to reduce passenger delays.

7 Conclusions and Future Research

In this paper, a dynamic passenger assignment model is proposed considering major disruptions that require trains to be cancelled or short-turned. Information

interventions are introduced by delivering the information of service variations and the information of train congestion at different locations. By applying the model on part of the Dutch railway network where a complete track blockage is assumed during one morning peak hour, it is found that when vehicle capacities are always sufficient (i.e. no denied boarding), publishing service variations at both stations and trains helps to reduce the travel time increase due to the disruption, while additionally publishing train congestion does not make any sense. When vehicle capacities are in short supply (i.e. denied boardings exist), additionally publishing train congestion can reduce the travel time increase due to the disruption, of which the performance depends on how a train is defined as highly congested in order to proactively avoid some passengers boarding the next run of the train.

Although only one case is performed in this paper, more applications could be performed with the proposed model. For example, considering the fluctuation of day-to-day passenger demand and the frequency of disruptions, reasonable vehicle capacity reservations for improving the service resilience during disruptions can be proposed. Besides, the proposed assignment model will be applied on larger networks and combined with rescheduling models in the future, where passengers will be grouped to speed the computation.

Acknowledgements 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.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Binder S, Maknoon Y, Bierlaire M (2017a) Exogenous priority rules for the capacitated passenger assignment problem. *Transp Res B Methodol* 105:19–42
- Binder S, Maknoon Y, Bierlaire M (2017b) The multi-objective railway timetable rescheduling problem. *Transport Res Part C Emerg Technol* 78:78–94
- Cats O, Jenelius E (2014) Dynamic vulnerability analysis of public transport networks: mitigation effects of real-time information. *Netw Spat Econ* 14(3–4):435–463
- Cats O, West J, Eliasson J (2016) A dynamic stochastic model for evaluating congestion and crowding effects in transit systems. *Transp Res B Methodol* 89:43–57
- Corman F, D'Ariano A, Marra AD, Pacciarelli D, Samà M (2017) Integrating train scheduling and delay management in real-time railway traffic control. *Transport Res Part E Logist Transport Rev* 105:213–239
- Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) Introduction to algorithms, 3rd. MIT Press, Cambridge
- 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*, pp 8–10
- Dollevoet T, Huisman D, Schmidt M, Schöbel A (2012) Delay management with rerouting of passengers. *Transp Sci* 46(1):74–89
- Gentile G, Noekel K (2016) Modelling public transport passenger flows in the era of intelligent transport systems. Springer, Cham

- Ghaemi N, Cats O, Goverde RMP (2017a) A microscopic model for optimal train short-turnings during complete blockages. *Transp Res B Methodol* 105:423–437
- Ghaemi N, Cats O, Goverde RMP (2017b) Railway disruption management challenges and possible solution directions. *Public Transport* 9(1–2):343–364
- Hamdouch Y, Lawphongpanich S (2008) Schedule-based transit assignment model with travel strategies and capacity constraints. *Transp Res B Methodol* 42(7):663–684
- Hamdouch Y, Szeto W, Jiang Y (2014) A new schedule-based transit assignment model with travel strategies and supply uncertainties. *Transp Res B Methodol* 67:35–67
- Kanai S, Shiina K, Harada S, Tomii N (2011) An optimal delay management algorithm from passengers' viewpoints considering the whole railway network. *J Rail Transp Plann Manage* 1(1):25–37
- Khani A, Hickman M, Noh H (2015) Trip-based path algorithms using the transit network hierarchy. *Netw Spat Econ* 15(3):635–653
- Larrain H, Muñoz JC (2008) Public transit corridor assignment assuming congestion due to passenger boarding and alighting. *Netw Spat Econ* 8(2):241–256
- Nuzzolo A, Russo F, Crisalli U (2001) A doubly dynamic schedule-based assignment model for transit networks. *Transp Sci* 35(3):268–285
- Poon M, Wong S, Tong C (2004) A dynamic schedule-based model for congested transit networks. *Transp Res B Methodol* 38(4):343–368
- Rückert R, Lemnian M, Blendinger C, Rechner S, Müller-Hannemann M (2017) PANDA: a software tool for improved train dispatching with focus on passenger flows. *Public Transp* 9(1–2):307–324
- Sato K, Tamura K, Tomii N (2013) A MIP-based timetable rescheduling formulation and algorithm minimizing further inconvenience to passengers. *J Rail Transp Plann Manage* 3(3):38–53
- Schöbel A (2001) A model for the delay management problem based on mixed-integer-programming. *Electron Notes Theor Comput Sci* 50(1):1–10
- Wardman M (2004) Public transport values of time. *Transp Policy* 11(4):363–377
- Zhu Y, Goverde RMP (2017) Dynamic passenger assignment during disruptions in railway systems. In: 5Th IEEE international conference on models and technologies for intelligent transportation systems (MT-ITS), Naples, 26–28 June 2017

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.