ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Train rescheduling for large-scale disruptions in a large-scale railway network

This is the final peer-reviewed author’s accepted manuscript (postprint) of the following publication:

Published Version:
Zhang C., Gao Y., Cacchiani V., Yang L., Gao Z. (2023). Train rescheduling for large-scale disruptions in a large-scale railway network. TRANSPORTATION RESEARCH PART B-METHODOLOGICAL, 174, 1-33 [10.1016/j.trb.2023.102786].

Availability:
This version is available at: https://hdl.handle.net/11585/956062 since: 2024-04-30

Published:
DOI: http://doi.org/10.1016/j.trb.2023.102786

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

14 July 2024

This is the final peer-reviewed accepted manuscript of:

Zhang, C., Gao, Y., Cacchiani, V., Yang, L., & Gao, Z. (2023). Train rescheduling for large-scale disruptions in a large-scale railway network. Transportation Research Part B: Methodological, 174, 102786.

The final published version is available online at: **https://doi.org/10.1016/j.trb.2023.102786**

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/)

When citing, please refer to the published version.

Train rescheduling for large-scale disruptions in a large-scale railway network

Chuntian Zhang[1], Yuan Gao[2*], Valentina Cacchiani[3], Lixing Yang[4], Ziyou Gao[1]

1. *School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China*

2. *School of Management and Economics, Beijing Institute of Technology, Beijing, 100081, China*

3. *DEI, University of Bologna, Viale Risorgimento 2, Bologna I-40136, Italy*

4. *State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University Beijing 100044, China*

Abstract

This paper studies the problem of rescheduling trains in a large-scale railway network with the characteristics of long distance, long time horizon and a large number of trains, where a disruption causes the paralysis of a significant part of the network for a long duration. As rescheduling measures we consider train reordering and retiming as well as the option of rerouting trains along alternative paths in the railway network. Although rerouting was not previously employed in large-scale long-distance networks, we show its benefit in reducing passenger delays. We formulate the problem as an integer linear programming (ILP) model on a space-time network with the goal of minimizing the total passenger delay and the number of passengers that could not reach their destination. In order to effectively solve the ILP model for real-world instances, we propose a heuristic algorithm (LR-H), based on the Lagrangian relaxation (LR) of a subset of constraints in the ILP model. LR allows decomposing the problem into a series of independent train-based subproblems which are easy to solve. Due to the large number of constraints and to cope with the real-time requirement, LR-H employs dynamic constraint-generation.

We tested LR-H on railway networks of different sizes under several disruption scenarios: first, we compared the results obtained by LR-H on a small-size instance with those found by the general-purpose ILP solver GUROBI, showing that LR-H can find near-optimal solutions in significantly shorter computing times. Then, we applied LR-H to a real-world instance of a railway network in China with 350 trains in different disruption scenarios. The experimental results show that, LR-H can obtain near-optimal solutions with an average optimality gap of 2.27% in an average computing time of about 300 seconds.

Keywords: Disruptions; Real-time train rescheduling; Integer linear programming; Lagrangian relaxation.

1 Introduction

In daily operations, disruptions may suddenly occur in a railway system due to several factors such as bad weather conditions, malfunctioning railway infrastructure, rolling stock breakdown. For instance, Jespersen-Groth et al. (2009) report that there are about 17 infrastructure-related disruptions in the Dutch railway network every day, with an average duration of 1.8 h. Zhu and Goverde (2017), by analyzing historical disruption data of the Netherlands railways, show that the unexpected disruptions occur in railways on a daily basis. Xu et al. (2016) report, based on data from the Chinese high-speed railway, that although the frequency of disruptions is relatively low, once a disruption occurs, it has a large impact on the running traffic. According to the statistics published by China Daily, on August 24, 2017, a sudden heavy rain in Wuhan caused the Wuhan station to be flooded, and normal operations were

[*] Corresponding Author: gaoyuan@bit.edu.cn

not restored after more than four hours. On May 1, 2021, due to the impact of strong wind on railway power supply, dozens of trains travelling on the Beijing-Guangzhou high-speed railway line were delayed for more than three hours, leading to a very poor travel experience for a large number of passengers.

When a disruption occurs, especially if it lasts several hours, a large number of trains have to deviate from their original timetable. Disruption management requires dispatchers to adjust the planned train schedules in real-time, by considering some strategies, such as reordering, retiming and rerouting, to ensure safety and reduce passenger dissatisfaction. The motivation of this research is to develop an effective algorithm to quickly generate solutions to the train rescheduling problem for large-scale disruptions. In particular, we focus on a large-scale railway network in China with a total length of 1935 km in which a disruption causes the complete blockage of part of the network (e.g., all tracks connected to an intermediate station are blocked), and consider duration of the disruption between two and six hours. A large-scale disruption in a large-scale railway network can affect many trains, especially if it lasts several hours, and these trains run for a long time period between their origin and destination stations: thus, the train rescheduling problem has to be solved by considering a long time horizon (Zhan et al. (2015)). Note that, due to the complexity of the problem, we do not integrate rolling stock circulation re-planning and crew rescheduling into the studied train rescheduling problem.

Rescheduling decisions found in the literature consist in train retiming and reordering, train service cancellations, short-turning (i.e., short-turn trains at a stations before the blocked segment to execute a train service in the opposite direction), flexible stopping and train rerouting. While short-turning is frequently used in European countries such as The Netherlands (Louwerse and Huisman (2014), Veelenturf et al. (2016), Ghaemi et al. (2017), Ghaemi et al. (2018a), Zhu and Goverde (2019)), a common strategy at Chinese railways is to stop the disrupted trains before the blocked segment until the disruption is solved (Zhan et al. (2015), Zhan et al. (2021)). This is mainly due to the seat reservation mechanism that does not allow passengers to directly travel on a different train than the booked one. However, if disrupted trains have to wait until the disruption is over, passengers can experience long delays. Existing works that consider long-distance networks employ train retiming, reordering and cancellation (Zhan et al. (2015), Zhan et al. (2016), Zhan et al. (2021)), but do not apply rerouting. In this paper, in addition to train retiming and reordering, we investigate the rerouting strategy in a large-scale network with the characteristics of long distance, long time horizon and a large number of trains to allow trains (that have not entered the blocked segment when the disruption occurs) to travel on alternative paths: in this way, trains can avoid the disrupted area, and passenger delays can be reduced.

Train rerouting involves the choice of changing the planned route and/or skipping some planned stops, and consequently some passengers may not reach their destination. Therefore, there is a trade-off between reducing passenger delays and increasing the number of abandoned passengers. Note that, we divide the trains into two types, namely the directly-affected trains and not-directly-affected trains, in which only the former trains may have the option of rerouting. A train is directly affected by a disruption, if the planned route of the train intersects the blocked sections during the blockage period, where the blocked section is the track or station blocked because of the disruption, and the duration of the disruption is the blockage period. In this setting, the experiment results show that, train rerouting turns out to effectively decrease passenger delays with a low percentage of abandoned passengers. In addition, all passengers who are not abandoned will stay on the booked train, thus respecting the seat reservation mechanism. In the next section, we give a more detailed overview of the literature on train rescheduling.

1.1 Literature review

The train rescheduling problem has attracted significant attention from railway operators and researchers, as shown in the recent surveys by Cacchiani et al. (2014), Fang et al. (2015), and Lamorgese et al. (2018). Compared to the train scheduling problem that is solved in planning, the train rescheduling problem requires to be solved in shorter computing times, and aims at reducing the inconvenience caused by disturbances and disruptions, often measured with respect to the planned schedule. According to the classification by Cacchiani et al. (2014), disturbances are relatively small perturbations, while disruptions are relatively large incidents. The former can usually be handled by adjusting the train timetable, while the latter sometimes requires changes both to the timetable and to the rolling stock plan and crew duties. To meet the real-time requirement, when dealing with disruptions, usually first the timetable is rescheduled and then the needed changes are applied to rolling stock circulation and crew duties (Dollevoet et al. (2016)). Therefore, different approaches have been proposed in the literature for dealing with disturbances or disruptions. We first briefly review, in Section 1.1.1, the literature related to train rescheduling under disturbances, and then analyze, in Section 1.1.2, works on train rescheduling under disruptions, that is the topic of our work. Finally, in Section 1.1.3 we report a comparison with the papers that are most related to our work.

1.1.1 Train rescheduling under disturbances

Most of the works in this category propose microscopic approaches, i.e., deal with a detailed description of the railway infrastructure. Many solution methods are based on formulating the problem as a job-shop scheduling problem by using the alternative graph model (Mascis and Pacciarelli (2002)). In particular, D'Ariano et al. (2007a), D'Ariano et al. (2007b), D'Ariano et al. (2008) and Corman et al. (2010) develop branch-and-bound algorithms and tabu search heuristics for train rescheduling, that are employed in a real-time traffic management system, called Railway traffic Optimization by Means of Alternative graphs (i.e., ROMA), to support controllers when handling disturbances. Some works focus on train rescheduling from the passenger perspective, i.e., how to determine a disposition timetable that minimizes the inconvenience for the passengers as much as possible. In particular, delay management consists of deciding if connecting trains should wait, thus guaranteeing the passengers connections, or should depart on time to reduce delays (Schöbel (2007), Schöbel (2009), Schachtebeck and Schöbel (2010)). Dollevoet et al. (2012) and Dollevoet et al. (2015) extend the delay management models by including passenger rerouting. Corman et al. (2017) propose a model for the integrated train scheduling and delay management to provide lower bounds for large instances, and develop heuristic algorithms based on decomposing the problem into a train scheduling problem and a passenger routing problem. Benders decomposition exact approaches are proposed by Lamorgese and Mannino (2015) and Lamorgese et al. (2016): in these approaches, the train rescheduling problem is decomposed into a line rescheduling problem and a set of station problems where train rerouting can be applied. A Mixed Integer Linear Programming (MILP) based heuristic, called RECIFE-MILP, is proposed by Pellegrini et al. (2014) to consider the infrastructure with fine granularity, and improved by Pellegrini et al. (2019) who introduced valid inequalities. A fast and effective parallel iterative heuristic algorithm, based on a space-time graph formulation, is proposed by Bettinelli et al. (2017): at each iteration, trains are scheduled following a given order, and then shaking methods are applied between consecutive iterations to consider different train orders. Gao et al. (2017) integrate the dynamic information of fault handling into the real-time automatic rescheduling strategy to deal with small disturbances. In dense railway systems, Altazin et al. (2020) combine a multi-objective optimization approach, that considers recovery time and the quality of service for passengers, with macroscopic simulation used to evaluate the passenger flows through the railway system. In single-track high-speed railway system, Zhang et al. (2021) propose a real-time train rescheduling strategy by considering disturbance uncertainties.

1.1.2 Train rescheduling under disruptions

Compared to disturbances, disruptions more deeply affect the train schedule, and require significant changes. Due to this additional complexity, most of the works propose macroscopic approaches, in which the railway infrastructure is represented at a high level. In addition, complete track blockages (i.e., disruptions that do not allow any train to run along the disrupted area) are often studied in the literature, since these disruptions impact on the train schedule more seriously than partial blockages.

The earliest works dealing with train rescheduling under disruptions were presented by Meng and Zhou (2011), Narayanaswami and Rangaraj (2013) and Louwerse and Huisman (2014)), and consider single or double track lines. In particular, Meng and Zhou (2011) consider a disrupted single-track rail line, and take into account random variations both in the running times and in the duration of the disruption. They propose a stochastic programming model and a multi-layer branching solution procedure that applies train retiming as a rescheduling measure. Narayanaswami and Rangaraj (2013) propose a MILP model that applies train retiming and reordering on a single-track railway line to minimize the total delay of trains measured at their destinations. The model is solved by a general-purpose solver on a small-scale artificial line. Louwerse and Huisman (2014) address both partial and complete blockages of a double track railway line. They develop an event-activity network ILP model that, in addition to train retiming and reordering, applies cancellation and short-turning. Constraints are imposed to respect the available number of rolling stock units, and to balance the number of non-canceled services in each direction. The model is tested on instances of the Dutch railways.

More recently, studies have focused on railway networks or long-distance lines with the aim of minimizing train delays and cancellations. Depending on the considered case study, different rescheduling measures are applied to reschedule trains. In particular, Zhan et al. (2015) consider a complete blockage of a long-distance high-speed railway line. They do not apply short-turning and rerouting: hence, trains have to wait outside the disrupted area until the disruption is solved. They develop an event-activity network MILP model that determines train reordering and retiming to be applied after the disruption is over, and train cancellation with the goal of minimizing the total weighted train delay and the number of canceled trains. The model is tested on the real data of Beijing-Shanghai high-speed line, by considering rescheduling the timetable for a whole day, since a large disruption on long-distance lines can influence all scheduled trains. Zhan et al. (2016) solve the train rescheduling problem on a double-track high-speed railway in case of a partial blockage, by considering that trains of both directions can share the same track, and employing retiming, reordering and cancellation. They consider three adjustment strategies used in practice, and propose three event-activity network MILP models to compare them. A rolling horizon approach is proposed in order to split the rescheduling problem into several stages (where at each stage, the information on the duration of the disruption is updated). The approach is tested on an instance of the Beijing-Shanghai high-speed line. Veelenturf et al. (2016) extend the work by Louwerse and Huisman (2014) to deal with complete and partial track blockages in a railway network, and to include train rerouting along different paths to avoid the disrupted area with the goal of minimizing the number of canceled train services and the train delays. Instances of the Dutch railways are tested under several disruption scenarios. Ghaemi et al. (2018a) consider a disruption causing a complete blockage, and propose a MILP model that allows for short-turning at two alternative stations for each train. They tested it on a dense double-track Dutch railway corridor showing that different disruption periods and lengths result in different optimal short-turning choices. A complete blockage in a metro line is considered by Wang et al. (2021), who propose a multi-objective MILP model for the problem of integrated train rescheduling and rolling stock circulation re-planning. The goals consist of minimizing deviations from the planned timetable, cancellations and headway deviations. A two-stage approach, that employs retiming, cancellation and short-turning, combined with a heuristic constraints filtering is proposed and tested on real-world instances of Beijing Subway lines.

As mentioned above, the majority of the works consider the infrastructure described at macroscopic level: an exception is the work by Ghaemi et al. (2017) who extend the MILP model by Pellegrini et al. (2014) to deal with complete blockages at microscopic level, and apply it on two corridors of the Dutch railway network including a large station with multiple routes and platform tracks. In addition, most works assume that the length of the disruption is provided: differently, Ghaemi et al. (2018b) and Zhu and Goverde (2020a) take into account the uncertainty of the disruption duration. The former proposes an iterative approach that starts with a duration prediction, and then updates the train schedule, based on a new duration prediction, while the latter develops a rolling horizon two-stage stochastic model. Both approaches are tested on a part of the Dutch railway network. Usually, due to the problem complexity, a single disruption at a time is considered: on the contrary, Zhu and Goverde (2021) develop a MILP model and a rolling horizon algorithm to deal with multiple connected disruptions (i.e., disruptions that occur at different geographic locations but have overlapping periods), and test them on instances of a Dutch network.

All the above mentioned papers consider the minimization of train delays and cancellations, however several works focus on passenger-oriented rescheduling, i.e., integrate passenger behavior or passenger rerouting in the problem or consider passenger demand as a weight of the impact of rescheduling decisions on the passenger inconvenience. In particular, Cadarso et al. (2013) consider a disruption in a rapid transit network, and propose a two-step approach for an integrated rescheduling of the timetable and rolling stock, taking into account the passenger demand behavior. In the first step, the approach computes the disrupted demand before adjusting the timetable, and, in the second step, solves a MILP model for the timetabling and rolling stock scheduling problem. The approach is tested on instances of the regional network in Madrid. Binder et al. (2017) formulate a multi-objective ILP model based on a space-time graph with integrated passenger rerouting (i.e., passengers adapt their travel routes based on the rescheduled train services). The aim is to minimize the generalized travel time of the passengers, the operational cost of running trains and the deviation from the undisrupted timetable. Rescheduling strategies consist of train reordering and retiming, cancellations, rerouting and adding emergency trains. The second and third objectives are included as ε-constraints, and the Pareto frontier is derived to evaluate the trade-off between the three goals. The approach is tested on a heavily utilized part of the Dutch railway network. Veelenturf et al. (2017) propose a heuristic iterative approach for timetable and rolling stock rescheduling, and passenger rerouting: in this case, timetable rescheduling is limited to cancellations and flexible stopping. The approach updates the timetable and the rolling stock schedule, and uses a deterministic simulation procedure to determine the passenger flows. The approach is tested on a busy part of the Dutch railway network. In Zhu and Goverde (2019), the goal accounts for the impact of train cancellations and delays on passengers by considering these rescheduling decisions as weighted according to the time-dependent passenger demand. In particular, the impact of the dispatching operations is measured in terms of passenger planned paths that are used as weights in the objective function. They extend the MILP model by Ghaemi et al. (2018a) by allowing for flexible stopping and more flexible short-turning strategies (i.e., a train can short-turn at one among all short-turn stations before the disrupted area). The model is applied on a Dutch railway network under several disruption scenarios. Integrated passenger rerouting and timetable rescheduling is studied by Zhu and Goverde (2020b) who propose an event-activity network MILP model, that includes the dispatching measures of re-timing, re-ordering, cancellation, flexible stopping and short-turnings, with the goal of minimizing the generalized passenger travel times. The model employs a transition network used to describe the alternative paths available to passengers. An Adapted Fix-and-Optimize algorithm, that iteratively solves the model with a limited number of passenger groups, is designed to solve the problem and tested on a part of the Dutch railways. Hong et al. (2021) formulate a bi-objective MILP model for the integrated train rescheduling and passenger reassignment to determine how many disrupted passengers should be assigned to each train, in order to deal with a seat reservation mechanism and limited train capacity. The two objectives are the maximization of the number of disrupted passengers reaching their destination and the minimization of the total weighted train delay. Retiming, reordering, rerouting (applied inside stations for platform assignment), and additional stops are used as rescheduling strategies. Pareto optimal solutions are determined by a weighted-sum approach tested on a section of the Beijing-Shanghai high-speed line. Zhan et al. (2021) propose an ILP model based on a space-time graph for the integrated train rescheduling and passenger rerouting problem. The objective is the minimization of the total passenger travel cost (accounting for passenger travel time and number of transfers) and the cost for operating trains. The model is decomposed into two subproblems dealing with train rescheduling and passenger rerouting, respectively, and is solved by an alternating direction method of multipliers algorithm that considers train reordering, retiming and cancellation. The algorithm is tested on a part of the Chinese high-speed railway network.

1.1.3 Comparison with the literature

Table 1 summarizes studies that are closely related to our work. In particular, we consider a railway network described at a macroscopic level of detail, and assume that the disruption duration is known. We compare the existing works based on three aspects: solution method, problem description (i.e., objective, train rescheduling measures, and consideration of passengers in the rescheduling process), and reported computational experiment scale. We divide the table into two parts: in the top part we report works that consider railway lines and in the bottom one those that consider railway networks.

Table 1: Comparison of closely related references

Publication	Solution method	Objective function	Rescheduling measure	Passengers	Largest instance size
Zhan et al. (2015)	CPLEX	Min. C & TD	RT,RO,C	N	Line (LD), S=23, T=42, D=2h, H=[8:00, 20:00]
Zhan et al. (2016)	CPLEX+ Rolling horizon	Min. C & Dev	RT,RO,C	N	Line (LD), S=23, T=180, D=2h, H=[8:00, 20:00]
Ghaemi et al. (2018a)	GUROBI	Min. C & TD	RT,C,S	N	Line, S=13, T=28 (per hour), D=5h
Hong et al. (2021)	CPLEX	Multi-obj-1	RT,RO,RR	R	Line (LD), S=8, T=20, D=55min, H=[19:19, 23:19]
Veelenturf et al. (2016)	CPLEX	Min. C & TD	RT,RO,C,S,RR	N	Network, S=26, T=165, D=2h, H=2.75h
Binder et al. (2017)	CPLEX+ε constraints	Multi-obj-2	RT,RO,C,RR	R	Network, S=11, T=24, D=2h, H=2h
Zhu and Goverde (2019)	GUROBI	Min. C & PD	RT,RO,C,S	S	Network, S=17, T=24 (per hour), D=3h, H=[7:00, 12:00]
Zhu and Goverde (2020b)	AFaO	Min. GPTT	RT,RO,C,S	R	Network, S=17, T=24 (per hour), D=2h, H=[7:00, 11:00]
Zhan et al. (2021)	ADMM	Min. PTC&TOC	RT,RO,C	R	Network (LD), S=17, T=20, D=2h, H=[10:00, 16:00]
This paper	Lagrangian heuristic	Min. C & PD	RT,RO,C,RR	S	Network (LD), S=47, T=350, D=6h, H=[6:00, 24:00]

Solution method: Adapted Fix-and-Optimize (AFaO); Alternating direction method of multipliers (ADMM).
Objective function: Cancellations (C); Train delays (TD); Deviations (Dev); Passenger delays (PD); Generalized passenger travel times (GPTT); Passenger travel cost and train operation cost (PTC&TOC);
Maximize the number of saved passengers and minimize the weighted total train delay (Multi-obj-1).
Minimize passenger inconvenience, operational cost and deviation from undisrupted timetable (Multi-obj-2);
Rescheduling measure: Retiming (RT); Reordering (RO); Cancelling (C); Short-turning (S); Rerouting (RR).
Passengers: Not considered (N); Static passenger demand (S); Passenger Rerouting (R).
Largest instance size: Line or network and if long-distance (LD); Number of stations (S); Number of trains (T); Disruption duration (D); Planning time horizon (H).

From Table 1, that summarizes our literature overview, we can observe that works dealing with large-scale long-distance networks do not apply train rerouting. The only exception is the model by Hong et al. (2021) that, however, focuses on a single direction line, and applies rerouting inside stations. In our work, rerouting is applied to select alternative paths in the railway network. As mentioned in Veelenturf et al. (2016), one advantage of having the option to reroute a train service is that passengers wishing to use this train service do not have to reroute themselves. Note that some studies (e.g., D'Ariano et al. (2019), Zhang et al. (2020) and Zhang et al. (2022)) on the integrated optimization of train scheduling and maintenance planning in railway networks also consider the selection of train routes. However, this integrated optimization problem belongs to the tactical level, and there are less strict requirements for computing time. Besides, the cancelling action in our work represents the train service cancellation, which consists only in cancelling some planned stops (since trains are rerouted and thus may skip a subset of stations), while all other train services are executed (possibly after the disruption is over). We consider a rescheduling problem in a network with a long time horizon, e.g., from the end of a disruption to the end of a day. As a result, there are enough alternative space-time routes for the involved trains, which is verified by our experiments. Since partial service cancellation can reduce the impact on the planned rolling stock schedule, we do not consider complete service cancellation. However, it is not difficult to include the option of train cancellation in our model by adding the candidate cancellation arcs for each train in the space-time network. Based on the above considerations, in Table 1, we also include the train cancelling strategy (i.e., train service cancellation) as one of the rescheduling measures.

Since we deal with a large-scale long-distance network, we have to consider a longer time horizon and a larger number of trains compared to most of the previous works, which make the considered problem challenging. We consider disruption durations between two and six hours: although the latter ones are less common, we want to evaluate the performance of the proposed algorithm in this complex case as it may occur in practice. As observed from the literature analysis, some recent works focus on passenger-oriented rescheduling: to keep the computing time short, we choose to employ a static passenger demand as in Zhu and Goverde (2019), i.e., we neglect the passenger rerouting but consider the impact that rescheduling decisions have on passengers by weighting cancellations and delays according to the number of affected passengers. Most of the works consider a time limit of 300 s, although longer computing times are sometimes required, especially when passenger rerouting is integrated with timetable rescheduling (Binder et al. (2017), Zhan et al. (2021)). In our computational experiments, we consider a time limit of 300 s, and we also evaluate the performance of the algorithm with shorter (1 min) and longer (10 min) computing times. Similar to Zhu and Goverde (2020b) and Zhan et al. (2021), due to the high problem complexity and the real-time requirement, we propose a customized algorithm, and directly apply a general-purpose solver to solve the proposed model only for comparison purposes.

1.2 Contributions

Disruption management is a very challenging task in railway operations, especially due to the real-time requirement, and is relevant from a practical perspective as disruptions deeply affect passenger satisfaction. We aim at defining a suitable model and developing an efficient heuristic algorithm to solve the train rescheduling problem in a railway network with the goal of reducing passenger inconvenience. The main contributions of this paper are as follows:

- We incorporate the rerouting strategy in train rescheduling in a large-scale railway network in which a large-scale disruption causes the complete blockage of part of the network (e.g., all tracks connected to an intermediate station are blocked). The large-scale railway network we study includes three main characteristics: long distance, long time horizon and a large number of trains. In previous studies, the rerouting strategy was not considered in such a large-scale network. We want to investigate how it can help reduce passenger delays. In a long-distance network, when a large-scale disruption occurs, more trains have to be considered in the rescheduling process and a longer planning time has to be taken into account, especially when rerouting is applied to trains that travel along alternative lines. These features make the problem more challenging.

- An efficient Lagrangian relaxation based heuristic algorithm (LR-H) is proposed to solve this Train Rescheduling Problem (TRP). LR-H is based on an ILP model formulated on a space-time network (with 1 min time discretization), in which track capacity and station capacity are also considered. The model minimizes the number of passengers that could not reach their destination, and the total delays of all passengers that reached their destination. LR-H applies Lagrangian relaxation of a subset of constraints in the ILP model to decompose the problem into a series of independent subproblems, and employs dynamic constraint-generation to handle the large number of constraints in the model. LR-H is compared with the general-purpose solver GUROBI on a small-size instance, and applied to solve a real-world instance of a railway network in China with 350 trains: the results show that, in most disruption scenarios, high quality solutions are obtained within 60 seconds of computing time, and improved optimality gaps (on average 2.27%) are achieved in an average computing time of about 300 seconds.

The remainder of the paper is organized as follows. Section 2 gives a detailed description of the considered problem and lists the relevant assumptions we make. Section 3 describes how we construct a space-time network used to formulate the proposed ILP model for the TRP, presented in Section 4. In Section 5, we present the proposed heuristic algorithm LR-H, based on Lagrangian relaxation, and, in Section 6, we report the results of the computational experiments we carried out on two railway networks with different size to verify the effectiveness of the method. Finally, some conclusions and future research directions are presented in Section 7.

2 Problem Description and Assumptions

To illustrate the considered TRP, we first provide an example in Section 2.1. Then, we list all assumptions in Section 2.2, and finally describe the TRP in Section 2.3.

2.1 Example

Figure 1 shows a railway network with three lines, including 7 stations and 8 sections, in which the running time between stations is marked with red numbers. Due to a large-scale disruption, station 3 and all connected sections (2-3), (3-6) and (3-4) are blocked. Assume that the disruption starts at time σ_1 and lasts until time σ_2. Consider three trains scheduled in this network, as shown in the figure (a black circle represents a train stop): according to the planned schedule, train 1 departs from station 1 to station 4 and stops at station 3, train 2 departs from station 1 to station 7 and stops at stations 3 and 6, and train 3 departs, in the opposite direction, from station 7 to station 1 and stops at stations 6, 3 and 2.

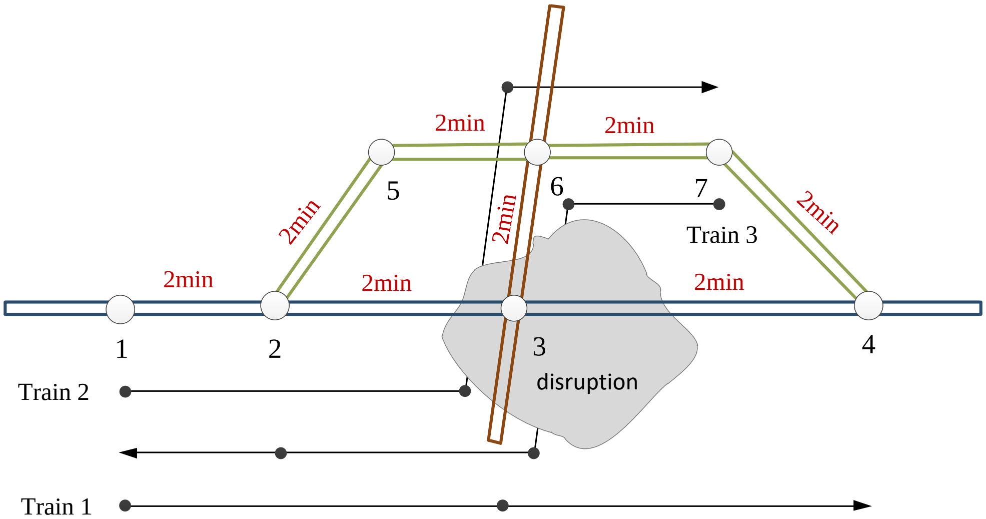


Figure 1: Example of a railway network with a disruption that affects station 3 and connected sections.

When the disruption occurs at station 3, it is necessary to quickly compute the rescheduled timetable for the three trains that run in the network. To obtain a feasible timetable, we consider four strategies: train retiming, train reordering, train service cancellation and train rerouting. However, we limit cancellation to those services that are not executed because of train rerouting. Therefore, when the disruption occurs, there are two choices for each train: one is to wait until the disruption is over and then continue on the original route, while the other is to choose an alternative route that avoids the disrupted area and possibly skips some stops, thus abandoning some passengers. Both alternatives then may require retiming and reordering. Clearly, if a train has already departed when the disruption occurs, then only the first option can be considered. For the other trains, it is necessary to determine which route has less impact on passengers taking into account the passenger delay and the number of abandoned passengers. In Figure 2, we show the planned (*Route k-Ori*) and alternative (*Route k-Re*) routes for each train k, where the disrupted station 3 is represented with a black circle.

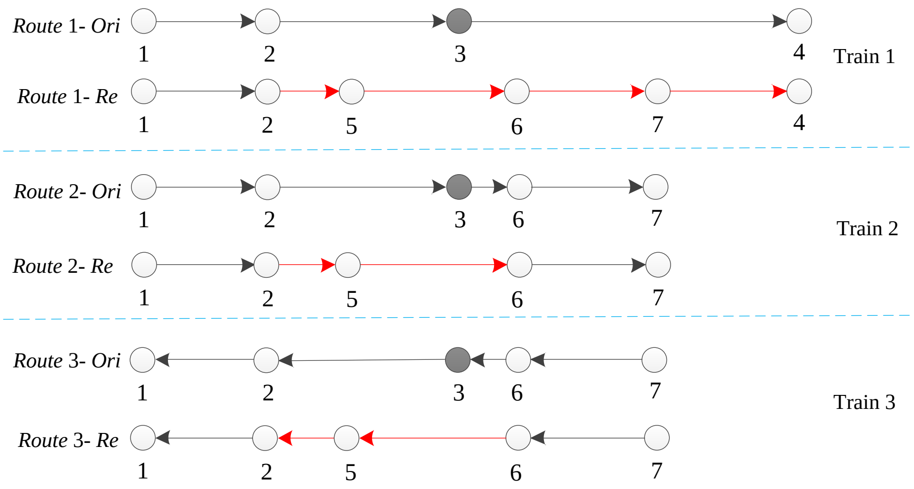


Figure 2: Planned and alternative train routes.

Assume that the disruption start time σ_1 and end time σ_2 are 5 and 12 respectively, and the departure time of trains 1, 2 and 3 from the origin station are 2, 4 and 2 respectively. Besides, assume that the number of passengers getting on and off each train at each station is 1 if the train has a planned stop at that station, and the penalty factor for abandoning each passenger is set to 10. We report the rescheduling strategies of the three trains in Table 2, where "Orig" represents the original schedule, "No-Rer" and "Rer", respectively, represent the strategy of not considering rerouting or allowing it. In particular, "No-Rer" corresponds to always selecting *Route k-Ori*, while "Rer" to performing either *Route k-Ori* or *Route k-Re* flexibly according to the actual situation. Column "Timetable" reports the sequence of visited stations according to the selected strategy and, in brackets, the corresponding arrival and departure times. For example, for the third train in the original schedule "Orig", $6(4,7)$ means that train 3 arrives at station 6 at 4 min and departs from station 6 at 7 min. In bold, we show the timetable changes with respect to the original schedule (retiming). In addition, f^* represents the sum of the passenger delays (including arrival and departure delays) and penalty for the abandoned passengers for each train, and F^* represents the sum of the total delay of all serviced passengers and the penalty for all abandoned passengers.

Table 2: Timetables under different rescheduling strategies.

Strategy	Train	Timetable	Choice	f^*	F^*
Orig	1	$1(2) \rightarrow 2(4,4) \rightarrow 3(6,8) \rightarrow 4(10)$	*Route* 1-*Ori*	–	–
	2	$1(4) \rightarrow 2(6,6) \rightarrow 3(8,10) \rightarrow 6(12,14) \rightarrow 7(16)$	*Route* 2-*Ori*	–	
	3	$7(2) \rightarrow 6(4,7) \rightarrow 3(9,11) \rightarrow 2(13,15) \rightarrow 1(17)$	*Route* 3-*Ori*	–	
No-Rer	1	$1(2) \rightarrow 2(4,4) \rightarrow \mathbf{3(13,15)} \rightarrow 4(17)$	*Route* 1-*Ori*	21	81
	2	$1(4) \rightarrow \mathbf{2(6,12)} \rightarrow 3(14,16) \rightarrow 6(18,20) \rightarrow 7(22)$	*Route* 2-*Ori*	30	
	3	$7(2) \rightarrow \mathbf{6(4,12)} \rightarrow 3(14,16) \rightarrow 2(18,20) \rightarrow 1(22)$	*Route* 3-*Ori*	30	
Rer	1	$1(2) \rightarrow 2(4,4) \rightarrow \mathbf{3(13,15)} \rightarrow 4(17)$	*Route* 1-*Ori*	21	61
	2	$1(4) \rightarrow \mathbf{2(6,6)} \rightarrow \mathbf{5(8,10)} \rightarrow 6(12,14) \rightarrow 7(16)$	*Route* 2-*Re*	20	
	3	$7(2) \rightarrow \mathbf{6(4,7)} \rightarrow \mathbf{5(9,11)} \rightarrow 2(13,15) \rightarrow 1(17)$	*Route* 3-*Re*	20	

We can see that, both in "No-Rer" and "Rer" cases, train 1 has to wait at station 3 since, when departing from station 2, it has not yet received information that station 3 is disrupted. Therefore, train 1 can only choose *Route* 1-*Ori* and temporarily stop at the track between stations 2 and 3 until the end of the disruption. For this train, there are no abandoned passengers since the route is unchanged, but passengers experience a delay of 21 minutes (7 minutes of delay for the passenger arriving at station 3, 7 for the passenger departing from station 3, and 7 for the passenger arriving at station 4). On the contrary, both trains 2 and 3 receive information on the disruption, hence their routes can be selected flexibly. If train 2 selects the alternative route ("Rer" strategy), then it will abandon two passengers (who want to arrive or depart from station 3) with a penalty of 20 but no delay: train services to and from station 3 are cancelled in this case. If instead it selects the original route ("No-Rer" strategy), no passenger will be abandoned, but there will be a delay of 30 minutes (no delay at station 2 because there was no planned stop, 6 minutes of delay for the passenger arriving at station 3, 6 for the passenger departing from station 3, 6 for the passenger arriving at station 6, 6 for the passenger departing from station 6, and 6 for the passenger arriving at station 7). A similar situation occurs for train 3. By comparing the strategies in terms of F^* that accounts both for delays and abandoned passengers, it is found that the rescheduling strategy considering rerouting of trains 2 ad 3 is preferred. Obviously, the best rescheduling strategy also depends on the penalty associated with abandoning passengers: in Section 6, we will compare the outcome with different penalty values.

2.2 Assumptions

We consider the following assumptions:

- Assumption 1. The duration of the disruption is known in advance.
- Assumption 2. Before the disruption occurs, all trains run normally according to the original timetable.
- Assumption 3. The trains that are already travelling towards the disrupted area when the disruption occurs have to wait at stations or tracks before the disrupted area until the end of the disruption.
- Assumption 4. The routes and stop plans of the not-directly-affected trains remain unchanged, but retiming and reordering can be applied to these trains (i.e., the arrival times, departure times and dwell times of the trains can be adjusted).
- Assumption 5. The rerouting strategy can only be employed when the rerouting requirements are respected. Specifically, the operation standards and signal systems of the lines before and after the rerouting must be consistent.
- Assumption 6. Once the disruption occurs, the trains can immediately receive the disruption information without any information delay.
- Assumption 7. Passenger reassignment is not considered: abandoned passengers (who are unable to reach their destination due to train rerouting) will get off at the last station before the train route changes.

As in many previous studies, we do not consider uncertainty of the disruption duration by Assumption 1. Assumptions 2-4 concern how the trains operate before and after the disruption occurs. In general, trains in the railway network operate regularly according to a given timetable before the disruption. After the disruption occurs, by Assumption 3 the trains, that are travelling towards the blockade area, cannot continue their trip, hence they stop before the disrupted area, since we do not consider short-turning, until the disruption is over for safety reasons. Considering that the railway dispatchers try to preserve the original plan of the trains, we especially make Assumption 4, which ensures that the routes of the not-directly-affected trains remain unchanged: rerouting is a major change that can cause skipping stops, hence it is only applied to trains that have to traverse the disrupted area. A not-directly-affected train is a train that, if it were the only train in the railway network, it would travel according to its original operation plan even during the disruption. In other words, it is a train whose planned route does not intersect the blocked sections during the blockage period. A not-directly-affected train may be rescheduled due to the knock-on effect of rescheduling the directly-affected trains. Assumption 5 specifies the requirements needed for adopting the rerouting strategy. Assumption 6 ensures that the disruption information is promptly transmitted to the train, so as to guarantee the safety of train operation.

Assumption 7 specifies that we do not consider passenger reassignment to different trains nor passenger rerouting: thus, when a train has to abandon some passengers, they will choose the nearest station to get off. Clearly, we account for minimizing the number of abandoned passengers, and aim at finding a rescheduled timetable in short computing time, so as to minimize the impact of disruption on passengers.

2.3 Problem description

We now describe the studied TRP, and introduce the used notation. Let H be the planning horizon, N the set of stations, L the set of tracks, and K the set of trains.

For each station $s \in N$, we are given its capacity B_s corresponding to the number of trains that can be simultaneously present at the station at any time. For each train $k \in K$, we are given the subset $\widehat{N}^k \subseteq N$ of stations that k visits, and the subset $\widehat{N}^k_d \subseteq N$ of stations where k has to stop according to the planned timetable. Moreover, for each train $k \in K$, we are given the minimum and maximum travel times between any pair of consecutive stations: in particular, let $r^k_{s_1,s_2}$ be the minimum travel time of train k from station s_1 to station s_2, and $R^k_{s_1,s_2}$ the maximum one ($(s_1, s_2) \in L$). In addition, for each station $s \in N$, we are given the minimum dwelling time ξ_s at the station, the minimum departure headway time h^d_s and the minimum arrival headway time h^a_s.

Since we deal with a rescheduling problem, we are also given the planned timetable, i.e., for each train $k \in K$, we know the scheduled departure time $\widehat{T}^d_{k,s}$ and the scheduled arrival time $\widehat{T}^a_{k,s}$ at each station $s \in \widehat{N}^k$. In addition, we have information on the disruption: we know the disruption duration $\mathcal{D}_t$, and the set $\mathcal{D}_s$ of tracks that are blocked. Based on the time and location of the disruption, we identify the subset $K_1 \subseteq K$ of trains it directly affects, and the subset $K_0 = K \setminus K_1$ of trains that do not travel along the disrupted area during the disruption period.

Finally, we are given the number of passengers booked on each train $k \in K$: in particular, let $q^{on}_{k,s}$ be the number of passengers who get on train k at station s, $q^{off}_{k,s}$ the number of passengers who get off train k at station s, and $Q_{k,s}$ their sum ($s \in \widehat{N}^k$).

Since the disruption blocks some tracks, the planned timetable is often no longer feasible, and a rescheduled timetable has to be determined. Clearly, the departure times of all trains cannot be earlier than in the planned timetable, in order to allow passengers to get on the trains. In addition, no train can travel along the disrupted area. We assume that trains already running towards the disrupted area when the disruption occurs can only wait at a station or track until the disruption is over. To reschedule trains we apply retiming, reordering and rerouting, but the latter strategy is only used for the directly-affected trains: indeed, by modifying the route, some stops may be skipped, thus causing abandoned passengers. Therefore, rerouting can cause the cancellation of some services. Since we deal with long-distance trains, retiming, reordering and rerouting can also affect trains running considerably later than the end of the disruption: therefore, we apply the rescheduling process until the end of the time horizon.

To obtain a feasible rescheduled timetable, track capacity constraints must be satisfied. In particular, the minimum headway time between consecutive departures of trains from the same station must be respected. Similarly, the minimum headway time between consecutive arrivals of trains at the same station must be respected. In addition, overtaking can only take place at stations and not along a track. For each train, travel times and dwell times must be satisfied. Finally, for each station, its capacity must be respected.

The goal is to minimize passenger departure and arrival delays, and, at the same time, minimize the number of passengers abandoned due to train rerouting. To better account for passengers inconvenience, we weight the terms in the objective according the number of affected (delayed or rerouted) passengers.

3 Space-time Network

A space-time network is an effective representation method to capture and analyze the spatial and temporal characteristics of dynamic traffic systems. It can combine the physical railway network with time-dependent train paths (timetables). We incorporate time-dependent train paths and disruption area and period into a space-time network that will be used to formulate an ILP model to find an effective train rescheduling strategy.

To construct the space-time network, we discretize the time horizon H (e.g., in minutes), and represent all stations in N and connections between them in L expanded in time. The space-time network is represented as an acyclic directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ and $\mathcal{A}$ represent, respectively, the sets of nodes and arcs. Node set $\mathcal{V}$ contains three types of nodes for each station $s \in N$: arrival nodes (that belong to set A^s), departure nodes (that belong to set D^s) and passing nodes (that belong to set P^s). An example for two stations is shown in Figure 3: A^1, D^1 and P^1 represent, respectively, the set of arrival nodes, departure nodes and passing nodes at station 1 (and similarly A^2, D^2 and P^2 for station 2). Nodes in P^1 are constructed for the trains that do not need to stop at station 1. Specifically, if a train does not stop at station 1, its path arrives at a node in P^1 and then departs from the node directly; if the train stops at station 1, its path must first arrive at a node in A^1, and then depart from a node in D^1. Besides, node set $\mathcal{V}$ also includes a dummy source node σ^k and dummy sink node τ^k for every train $k \in K$.

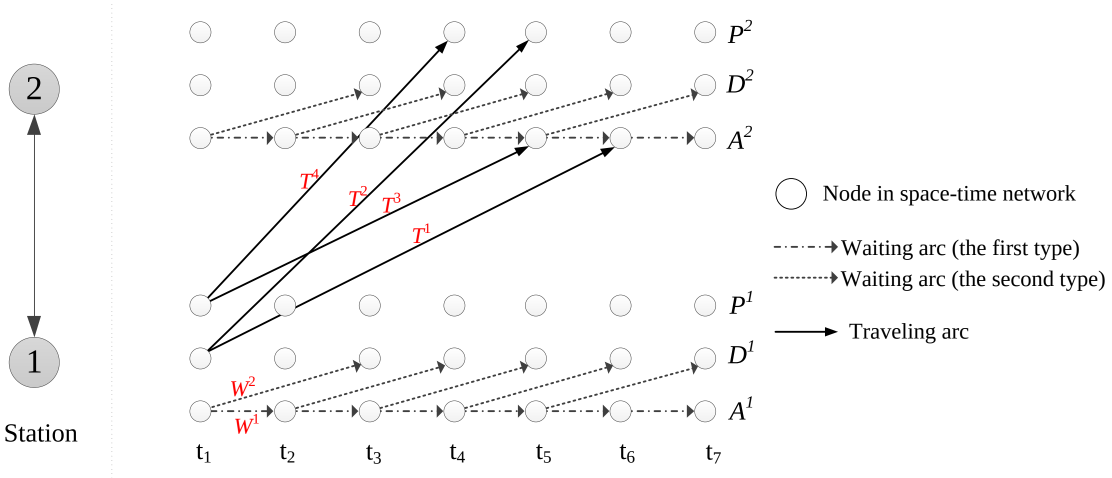


Figure 3: Illustration of the waiting arc and traveling arc in the space-time network

Arc set $\mathcal{A}$ contains four categories of arcs: waiting arcs, traveling arcs, disruption arcs and dummy arcs. Figure 3 shows an example of waiting and traveling arcs. We distinguish two types of waiting arcs, represented by dashed arrows in the figure: arcs of the first type (e.g., W^1) connect two different arrival nodes of the same station and represent a stop longer than the minimum dwelling time, while arcs of the second type (e.g., W^2) connect an arrival node and a departure node of the same station, and are used to ensure the minimum dwelling time of the train at the station. In addition, we distinguish four types of traveling arcs, represented by solid arrows in the figure, based on whether the train path stops or not at the starting and ending stations of the arc: T^1 is the traveling arc of trains that stop both at stations 1 and 2, while T^4 is the traveling arc of trains that do not stop at any of the two stations; T^2 is the traveling arc of trains that stop at station 1 but do not stop at station 2, while T^3 is the traveling arc of trains that do not stop at station 1 but stop at station 2. By considering different types of arcs and passing nodes P^s, we can obtain two advantages: (1) we can integrate the acceleration and deceleration times when the train stops at a station into the model; (2) we can ensure that the minimum dwelling time constraints of the trains can be satisfied by constructing as few arcs as possible. Waiting arcs at station s are denoted by $\mathcal{A}_a^s$ and those of the first type by $\mathcal{A}_{a'}^s$. Traveling arcs are indicated by $\mathcal{A}_t$, while outgoing arcs associated with the departure nodes of station s by $\mathcal{A}_d^s$. Note that we only introduce the definition of the set of arcs directly used in the ILP model to limit heavy notation (the definition of the set of arcs that do not appear in the model is not explicitly provided). Besides, disruption arcs, represented by $\mathcal{A}_{\mathcal{D}}$, correspond to all arcs contained in the disrupted area: they are used to identify waiting and traveling arcs that cannot be selected in any train path. Finally, dummy arcs connect the dummy source node and the dummy sink node, respectively, to all departure nodes of the train from its origin station, and to all arrival nodes of the train at its destination station.

Each train $k \in K$ visits a subset $\widehat{N}^k \subseteq N$ of the stations and has specific travel and dwell times. Therefore, we define $\mathcal{V}^k \subseteq \mathcal{V}$ and $\mathcal{A}^k \subseteq \mathcal{A}$, respectively, as the set of nodes and arcs of train $k \in K$. We show, in Figure 4, three paths of the space-time network, corresponding to the original and alternative routes of train 2 reported in Table 2 (for the railway network of Figure 1).

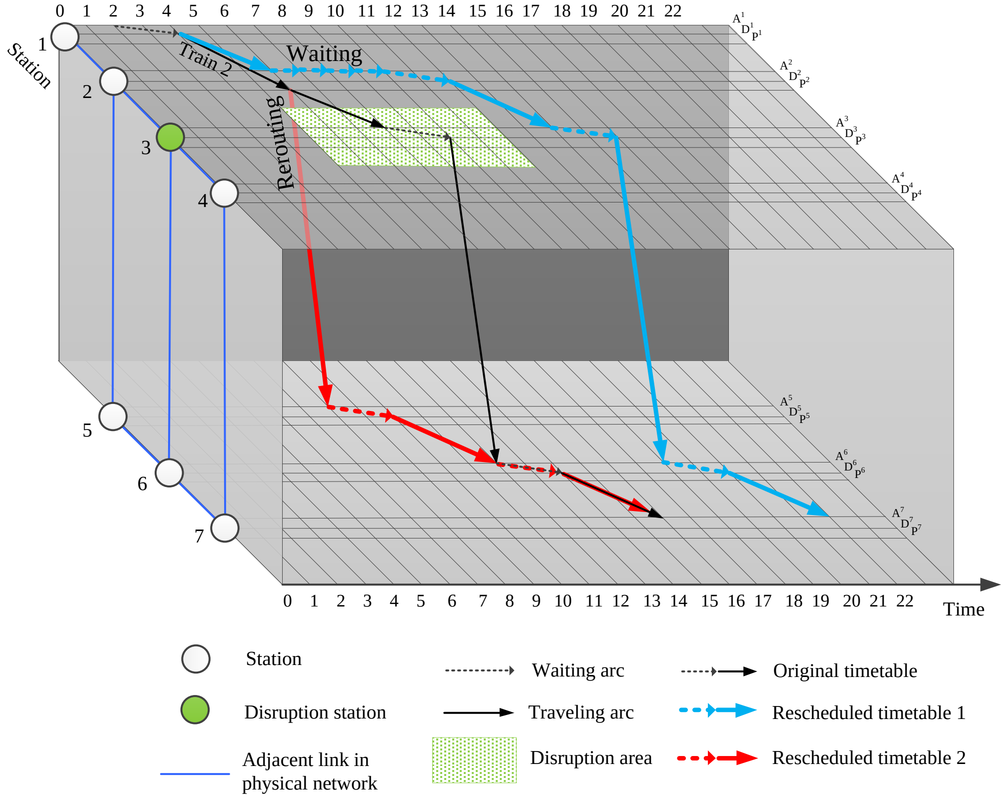


Figure 4: Space-time network of the given railway network

In Figure 4, we show the seven stations of the railway network, and the discretized time horizon with instants from 0 to 22 (i.e., all time instants used in Table 2). The disrupted area during the disrupted time period is shown in light green. The original timetable of train 2, that traverses the disrupted area, is represented by black lines, while we use blue and red lines to depict the alternative timetables of train 2 that avoid the disrupted area. In particular, the Rescheduled timetable 1 in blue represents the no rerouting strategy of Table 2, while the Rescheduled timetable 2 in red corresponds to the rerouting one. In the former case, train 2 waits until the disruption is over to continue running according to the original route, while in the latter one it selects another route that abandons the passengers of station 3, but reaches station 6 at the same time as in the original timetable (the black and the red lines coincide). We only reported two alternative timetables (paths) for train 2, but, obviously, the space-time network can contain a very large number of time-dependent paths for each train.

4 Model Formulation

4.1 Notation

The notation used in this paper is listed in Tables 3 and 4. In Table 3, we report the notation used for the given data, and in Table 4, we report the parameters used to formulate the ILP model (and its Lagrangian relaxation).

Table 3: Notation used for the given data

Notations	Definition
H	planning time horizon.
N	set of stations.
L	set of tracks.
K_0	set of the not-directly-affected trains.
K_1	set of the directly-affected trains.
K	set of all trains, $K = K_0 \bigcup K_1$.
B_s	maximum capacity of station s.
$\widehat{N}_d^k$	set of stations where train k is scheduled to stop in the original timetable.
$\widehat{N}^k$	set of all stations that train k visits in the original timetable.
$\widehat{T}_{k,s}^d$	departure time of train k from station s in the original timetable.
$\widehat{T}_{k,s}^a$	arrival time of train k at station s in the original timetable.
r_{s_1,s_2}^k	minimum travel time of train k between s_1 and s_2.
R_{s_1,s_2}^k	maximum travel time of train k between s_1 and s_2.
ξ_s	minimum dwelling time of each train at station s.
h_s^d	minimum headway between consecutive departures from station s.
h_s^a	minimum headway between consecutive arrivals at station s.
$\mathcal{D}_t$	disruption period.
$\mathcal{D}_s$	disruption location (set of blocked tracks).
$q_{k,s}^{off}$	number of passengers booked on train k that need to get off at station s.
$q_{k,s}^{on}$	number of passengers booked on train k that need to get on at station s.
$Q_{k,s}$	total number of passengers booked on train k that need to get off and on at station s, $Q_{k,s} = q_{k,s}^{off} + q_{k,s}^{on}$.

Table 4: Notation used in the formulation

Notations	Definition
$\mathcal{V}$	set of nodes.
$\mathcal{A}$	set of arcs.
$\mathcal{V}^k$	set of nodes of train k.
$\mathcal{A}^k$	set of arcs of train k.
i, j	index of nodes, $i, j \in \mathcal{V}$.
k	index of trains, $k \in K$.
s	index of stations, $s \in N$.
A^s	set of arrival nodes at station s.
D^s	set of departure nodes from station s.
P^s	set of passing nodes through station s.
$\mathcal{A}_t$	set of traveling arcs.
$\mathcal{A}_a^s$	set of waiting arcs of station s.
$\mathcal{A}_{a'}^s$	set of the first type of waiting arcs of station s.
$\mathcal{A}_d^s$	set of outgoing arcs associated with the departure nodes of station s.
$\mathcal{A}^s$	set of outgoing arcs associated with the nodes of station s.
$\mathcal{A}_\mathcal{D}$	set of arcs on the disrupted area.
σ^k/τ^k	dummy origin/destination nodes for train k.
$\theta(i)$	time instant associated with a given node $i \in \mathcal{V}$.
$C_{i,j}^k$	Lagrangian cost of arc (i, j) for train k.
δ	penalty factor of abandoning a passenger.

To model the TRP, we introduce the following binary decision variables:

$$y_{ij}^k \quad = 1 \text{ if train } k \text{ uses arc } (i,j), \qquad k \in K,\ (i,j) \in \mathcal{A}^k \ (= 0 \text{ otherwise}).$$

For each train $k \in K$ and each arc $(i,j) \in \mathcal{A}^k$, y_{ij}^k is equal to 1 if and only if the path of train k contains arc (i,j). These variables identify the time-dependent train paths (timetables) in the space-time network.

4.2 Constraints

In this section, we present the model constraints divided into categories according to their scope.

4.2.1 Flow balance constraints

They are used to ensure the feasibility of every time-dependent path of each train $k \in K$, that starts from the dummy source node σ^k and ends at the dummy sink node τ^k, as follows:

$$\sum_{(i,j)\in\mathcal{A}^k} y_{ij}^k - \sum_{(j,i)\in\mathcal{A}^k} y_{ji}^k = \begin{cases} 1, & i = \sigma^k \\ -1, & i = \tau^k \quad \forall k \in K \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

These constraints implicitly impose to respect, for each train $k \in K$, its travel time between each pair of visited stations and its minimum dwelling time at all visited stations, since only arcs in $\mathcal{A}^k$ can be selected for train k. Note that, since we distinguish between departure nodes and passing nodes in the space-time network, the flow conservation constraints guarantee that acceleration and deceleration times are correctly considered in the travel time, based on whether the train stops or not at a station.

4.2.2 Disruption related constraints

These constraints are required to deal with the disruption. There are four classes of constraints.

Departure time constraints They are used to ensure that, for each train $k \in K$ and each station $s \in \widehat{N}_d^k$ where the train was planned to stop, the actual departure time of k from s is not earlier than the departure time $\widehat{T}_{k,s}^d$ in the planned timetable.

$$\sum_{(i,j)\in\mathcal{A}_d^s, \theta(i)<\widehat{T}_{k,s}^d} y_{ij}^k = 0, \qquad \forall k \in K, s \in \widehat{N}_d^k \tag{2}$$

In particular, constraints (2) require that no arc (i,j) with a departure time instant $\theta(i)$ from station s before the planned one $\widehat{T}_{k,s}^d$ is selected for train k.

Route unchanged constraints To ensure that, for each not-directly-affected train $k \in K_0$, the route remains unchanged, the following constraints are imposed:

$$\sum_{(i,j)\in\mathcal{A}^s} y_{ij}^k \geq 1, \qquad \forall k \in K_0, s \in \widehat{N}^k \tag{3}$$

Constraints (3) require train k to select at least one outgoing arc among all arcs $\mathcal{A}^s$ from the nodes corresponding to station $s \in \widehat{N}^k$, so as to ensure the consistency of train passing route. Recall that $\widehat{N}^k$ represents the set of stations that train k visits in the original timetable.

Stop plan constraints To ensure that, for each train $k \in K_0$, the original stop plan is not modified, the following constraints are imposed:

$$\sum_{(i,j)\in\mathcal{A}_d^s} y_{ij}^k = 1, \qquad \forall k \in K_0, s \in \widehat{N}_d^k \tag{4}$$

In particular, if train k was planned to stop at station s in the planned timetable, then k will stop at s in the rescheduled timetable as well: this is imposed by requiring that train k selects exactly one arc from departure nodes corresponding to station $s \in \widehat{N}_d^k$.

Disrupted area constraints When an area is blocked by a disruption, no train can enter it, i.e., none of the arcs traversing the disrupted area can be selected:

$$y_{ij}^k = 0, \qquad \forall k \in K, (i,j) \in \mathcal{A}_{\mathcal{D}} \tag{5}$$

4.2.3 Conflict-free constraints

These constraints guarantee that a conflict-free timetable is determined. There are two classes of constraints.

Train-incompatibility constraints The track capacity constraints are imposed to respect the minimum headway times between train departures or arrivals, and to avoid train overtaking between consecutive stations along a line. They are formulated by introducing *incompatible arc sets* as proposed by Caprara et al. (2002). For clarity, we take two arcs $(i^{'}, j^{'})$ and (i, j) as an example to illustrate the definition of *incompatibility*. To ensure the safety of train operations, we must ensure that certain pairs of arcs associated with different trains cannot be simultaneously selected in a feasible solution. Therefore, for each pair of trains k and $k^{'}$, the two arcs $(i^{'}, j^{'})$ and (i, j) can not be simultaneously selected if at least one of the following conditions holds:

- the time instant of node $i^{'}$ precedes that of node i (i.e, $i^{'} \prec i$), and the time interval between the time instants associated with $i^{'}$ and i is smaller than h_s^d (i.e., $\theta(i)$-$\theta(i^{'})$<h_s^d); see Figure 5(a).
- the time instant of node $j^{'}$ precedes that of node j (i.e, $j^{'} \prec j$), and the time interval between the time instants associated with $j^{'}$ and j is smaller than h_{s+1}^a (i.e., $\theta(j)$-$\theta(j^{'})$<h_{s+1}^a); see Figure 5(b).
- the two arcs cross each other (i.e., $i^{'} \prec i$ and $j \prec j^{'}$); see Figure 5(c).

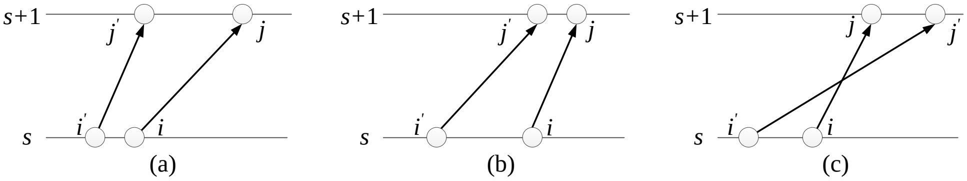


Figure 5: Incompatible pairs of arcs.

Such a pair of arcs $(i^{'}, j^{'})$ and (i, j) is called incompatible, thus the set consisting of all arcs that are incompatible with arc $(i^{'}, j^{'})$ is called the incompatible arc set associated with arc $(i^{'}, j^{'})$. As an example, Figure 6 further shows the departure incompatible arc set corresponding to travel arc $(i^{'}, j^{'})$: these arcs are incompatible due to the violation of the minimum departure headway time h_1^d from station 1.

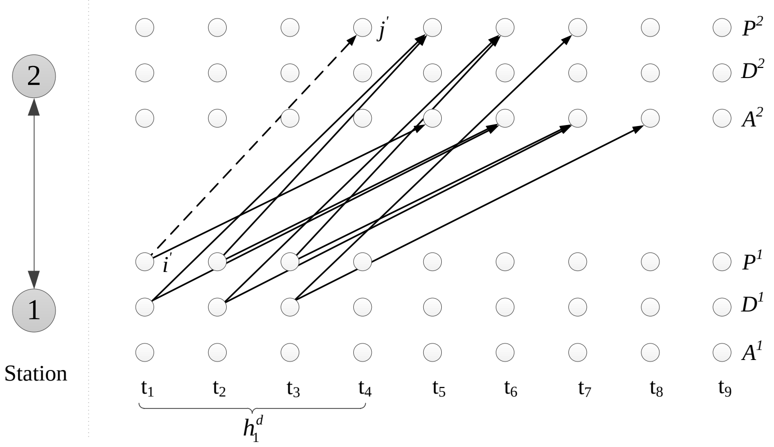


Figure 6: Illustration of the departure headway constraints.

In particular, all arcs representing departures from station 1 (even without a stop at station 1) that occur at a time instant between t_1 (the time instant of node i') and t_3 (with $t_3 = t_1 + h_1^d - 1$) are incompatible because they do not respect the minimum departure headway time. Hence, at most one arc can be selected from this set. Similarly, incompatible arc sets are identified due to violation of the minimum arrival headway time or of the overtaking constraints.

For each travel arc $(i^{'}, j^{'})$, we construct an incompatible arc set $\Psi(i^{'}, j^{'})$ containing all arcs that are in conflict with arc $(i^{'}, j^{'})$. As a result, the train-incompatibility constraints are formulated as:

$$\sum_{k \in K} \sum_{(i,j) \in \Psi(i^{'}, j^{'})} y_{ij}^k \leq 1, \qquad \forall (i^{'}, j^{'}) \in \mathcal{A}_t \tag{6}$$

Station capacity constraints At any time, each station $s \in N$ cannot accommodate a number of trains exceeding the station capacity limit B_s. The capacity B_s includes only the platforms for trains that stop at station s: indeed, trains passing through the station use another track, and, due to the headway constraints (6), no more than one of these trains can be present at the station at the same time. Therefore, station capacity constraints are referred to trains stopping at the station. To express these constraints, we introduce, for each waiting arc $(i', j') \in \mathcal{A}_{a'}^s$, the set $\varphi(i^{'}, j^{'})$ of waiting arcs in station s crossing time instant $\theta(i^{'})$. An example is shown by the arcs plotted in Figure 7, where, at time instant $\theta(i^{'})$, three trains are present at station s: two trains stop for the minimum dwelling time, while another train stops for an additional time instant (dashed arc $(i', j') \in \mathcal{A}_{a'}^s$). If the capacity B_s of station s is two, then the trains that choose the three arcs in set $\varphi(i^{'}, j^{'})$ at the same time are in conflict.

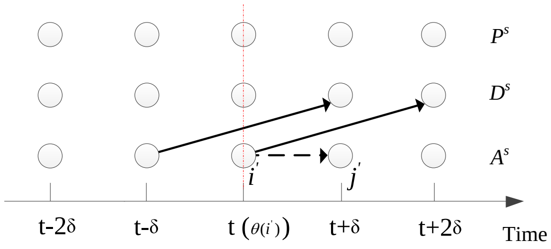


Figure 7: Illustration of the set of waiting arcs at time t.

Based on the introduced sets $\varphi(i^{'}, j^{'})$, the station capacity constraints can be formulated as:

$$\sum_{k \in K} \sum_{(i,j) \in \varphi(i^{'}, j^{'})} y_{ij}^k \leq B_s, \qquad \forall (i^{'}, j^{'}) \in \mathcal{A}_{a'}^s, \quad s \in N \tag{7}$$

Note that we only have to impose these constraints for the waiting arcs of the first type, i.e., those in set $\mathcal{A}^s_{a'}$, since waiting arcs of the second type (corresponding to the minimum dwelling times) will always appear in some set $\varphi(i^{'}, j^{'})$ of a waiting arc $(i^{'}, j^{'})$ of the first type.

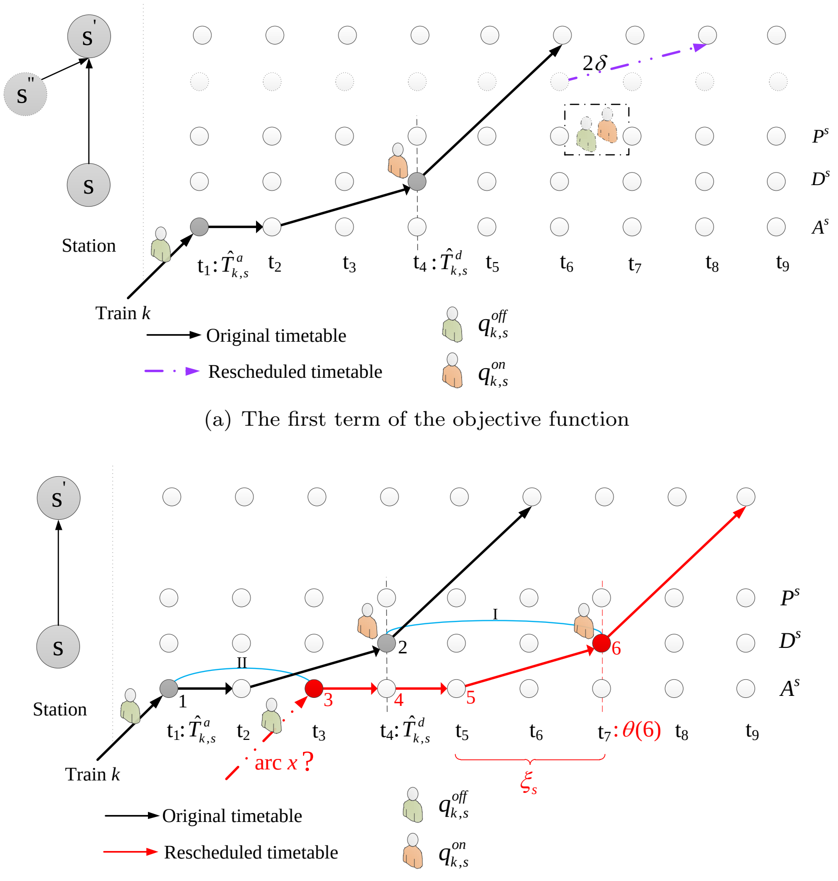
4.3 Objective function

When a disruption occurs, the goal is, on one hand, to limit the delays of passengers as much as possible and, on the other hand, to minimize the number of abandoned passengers, because train rerouting can make some passengers not able to reach their destination. We consider these two objectives in a weighted sum, where parameter δ is used to weight the respective importance of the two terms. In order to account for the passenger inconvenience, we consider the delay weighted by the static passenger demand: for each station s and train k, we consider the departure delay weighted by the number $q^{on}_{k,s}$ of passengers that get on train k at station s, and the arrival delay weighted by the number $q^{off}_{k,s}$ of passengers that get off train k at station s. The objective function is formulated in the following way:

$$\begin{aligned} F_0 = & \sum_{k \in K} \sum_{s \in \widehat{N}^k_d} \delta \times Q_{k,s} \times \Big(1 - \sum_{(i,j) \in \mathcal{A}^s_d} y^k_{ij}\Big) \\ & + \sum_{k \in K} \sum_{s \in \widehat{N}^k_d} q^{on}_{k,s} \times \Big[\sum_{(i,j) \in \mathcal{A}^s_d} y^k_{ij} \times \big(\theta(i) - \widehat{T}^d_{k,s}\big)\Big] \\ & + \sum_{k \in K} \sum_{s \in \widehat{N}^k_d} q^{off}_{k,s} \times \Big[\sum_{(i,j) \in \mathcal{A}^s_d} y^k_{ij} \times \big(\theta(i) - \xi_s - \widehat{T}^a_{k,s}\big) - \sum_{(i,j) \in \mathcal{A}^s_{a'}} y^k_{ij}\Big] \end{aligned} \tag{8}$$

The first term in the objective function (8) represents the penalty for abandoned passengers due to the selection of alternative routes for some trains. In particular, if no departure arc in $\mathcal{A}^s_d$ is selected for train k from station s, i.e., $\sum_{(i,j) \in \mathcal{A}^s_d} y^k_{ij} = 0$, then the passengers booked on that train cannot get on and off train k at station s. Hence, we consider as penalty $\delta Q_{k,s}$, where $Q_{k,s}$ is the number of abandoned passengers. For clarity, we report, in Figure 8, an example of objective function computation for train k scheduled to stop at station s, in which the planned timetable is shown in black. Two passengers book on train k: at station s, the green passenger gets off while the orange passenger gets on. In particular, Figure 8(a) shows a situation in which the rescheduled timetable (shown in purple) does not respect the original train stop plan. Indeed, it is obtained by rerouting train k that skips station s and travels through station $s^{''}$ (whose nodes are indicated with lighter color in the figure). Hence, both passengers are abandoned with total penalty 2δ.

The second term in the objective function (8) represents the departure delay of passengers who are not abandoned, i.e., $\sum_{(i,j) \in \mathcal{A}^s_d} y^k_{ij} = 1$. This term considers, for train k and station $s \in \widehat{N}^k_d$, the delay given by the difference between the actual departure time $\theta(i)$ of the selected departure arc $(i, j) \in \mathcal{A}^s_d$ and the planned departure time $\widehat{T}^d_{k,s}$, and weights the delay by the number $q^{on}_{k,s}$ of passengers departing from station s on train k. Figure 8(b) shows that the train stop plan is not modified, but the arrival and departure times of the train at station s are adjusted. Specifically, in the red rescheduled timetable, train k stops at station s but retiming is applied: the orange passenger has to depart at time $\theta(6) = t_7$ while the planned departure time was $\widehat{T}^d_{k,s} = t_4$, hence experiences a delay of 3 time instants (i.e., the part corresponding to the blue curve I).



(a) The first term of the objective function

(b) The second and third terms of the objective function

Figure 8: Illustration of objective function

The third term in the objective function (8) represents the arrival delay of passengers who are not abandoned, i.e., $\sum_{(i,j)\in\mathcal{A}_d^s} y_{ij}^k = 1$. This term considers, for train k and station $s \in \widehat{N}_d^k$, the delay given by the difference between the actual arrival time of train k and its planned arrival time $\widehat{T}_{k,s}^a$, and weights the delay by the number $q_{k,s}^{off}$ of passengers arriving at station s on train k. The part corresponding to the blue curve II in Figure 8(b) shows the arrival delay of green passenger. Note that the arrival arc x is an incoming arc of an arrival node at station s, which is not explicitly defined in Table 4. For the sake of notation simplicity, we deduce the calculation formula of the actual arrival time as a function of the departure arcs and waiting arcs at station s. Obviously, the arrival time associated with arrival arc x in Figure 8(b) is equivalent to $(\theta(6)$ - $\xi_s)$ - $(y_{4,5}^k + y_{3,4}^k) = t_3$, so the arrival delay experienced by the green passenger is t_3 - $\widehat{T}_{k,s}^a$(i.e., t_1) $= 2$ time instants. According to the definition of departure arc and waiting arc, we have $\sum_{(i,j)\in\mathcal{A}_{a'}^s} y_{ij}^k = 0$ if $\sum_{(i,j)\in\mathcal{A}_d^s} y_{ij}^k = 0$, which implies that the dwelling time is 0 if train k does not stop at station s. Therefore, the arrival delay can be expressed in the third term of objective function (8). We mention that the passenger arrival delay at the last station of each train $k \in K$ is taken into account by considering the arcs that connect nodes of the last station to the sink node τ^k, as a special case of departure nodes.

Note that we consider both departure and arrival delays in the objective function mainly for the following reasons: (1) we consider the double dissatisfaction of passengers. In the journey of passengers, we find that, in addition to the dissatisfaction caused by train arrival delay, departure delay can also cause passengers' anxiety; (2) considering both aspects of delay makes the model more general: if the second term in the objective function is removed, the problem only considers minimizing the arrival delay.

The model for TRP can be summarized as follows:

$$\begin{cases} \min F_0 \\ s.t. \\ \quad \text{Constraints } (1) - (7) \\ \quad y_{ij}^k \in \{0,1\}, \forall k \in K, (i,j) \in \mathcal{A}^k. \end{cases} \tag{9}$$

5 Solution method

In this section, we present the developed Lagrangian relaxation based heuristic algorithm (LR-H). In particular, we relax, in a Lagrangian way (Fisher (1981)), the train-incompatibility constraints (6) and the station capacity constraints (7) by incorporating them into the objective function with non-negative multipliers, respectively, λ_{ij} with $(i,j) \in \mathcal{A}_t$, and ρ_{ij} with $s \in N$ and $(i,j) \in \mathcal{A}_{a'}^s$. By applying this relaxation, the problem decomposes into $|K|$ train-specific subproblems. For each train $k \in K$, the subproblem consists in determining the time-dependent path in the space-time network $\mathcal{G}$ from σ^k to τ^k that satisfies all disruption related constraints and has the smallest Lagrangian cost: the latter accounts for the penalty due to the abandoned passengers, the passenger departure and arrival delays, and the Lagrangian penalties due to the violation of the relaxed constraints. In Section 5.1, we report the relaxed model and its decomposition into subproblems, and in Section 5.2, we describe the LR-H algorithm.

5.1 Lagrangian-relaxed model and decomposition

By relaxing, in a Lagrangian way, constraints (6) and (7), the Lagrangian-relaxed model (LRM) reads as follows:

$$\min F_1 = F_0 + \sum_{(i',j') \in \mathcal{A}_t} \lambda_{i'j'} \times \Big(\sum_{k \in K} \sum_{(i,j) \in \Psi(i',j')} y_{ij}^k - 1 \Big) + \sum_{s \in N} \sum_{(i',j') \in \mathcal{A}_{a'}^s} \rho_{i'j'} \times \Big(\sum_{k \in K} \sum_{(i,j) \in \varphi(i',j')} y_{ij}^k - B_s \Big)$$

$$\begin{aligned} & s.t. \\ & \text{Constraints } (1) - (5) \\ \lambda_{i'j'} &\geq 0, && \forall (i',j') \in \mathcal{A}_t \\ \rho_{i'j'} &\geq 0, && \forall s \in N,\ (i',j') \in \mathcal{A}_{a'}^s. \end{aligned}$$

Since all constraints that involve more than one train are relaxed, LRM can be decomposed into $|K|$ train-specific subproblems. The objective function F_1^k of each subproblem ($k \in K$) is reported in (10), where we explicitly write F_0 as in (8), and group terms related to the same arc type:

$$\begin{aligned} F_1^k = & \sum_{s \in \widehat{N}_d^k} \sum_{(i,j) \in \mathcal{A}_d^s} \Big[q_{k,s}^{on} \times \big(\theta(i) - \widehat{T}_{k,s}^d\big) + q_{k,s}^{off} \times \big(\theta(i) - \xi_s - \widehat{T}_{k,s}^a\big) - \delta \times Q_{k,s} \Big] \times y_{ij}^k \\ & - \sum_{s \in \widehat{N}_d^k} \sum_{(i,j) \in \mathcal{A}_{a'}^s} q_{k,s}^{off} \times y_{ij}^k + \sum_{s \in \widehat{N}_d^k} \delta \times Q_{k,s} + \sum_{(i',j') \in \mathcal{A}_t} \lambda_{i'j'} \times \Big(\sum_{(i,j) \in \Psi(i',j')} y_{ij}^k - 1 \Big) \\ & + \sum_{s \in N} \sum_{(i',j') \in \mathcal{A}_{a'}^s} \rho_{i'j'} \times \Big(\sum_{(i,j) \in \varphi(i',j')} y_{ij}^k - B_s \Big) \\ = & \sum_{(i,j) \in \mathcal{A}} C_{ij}^k \times y_{ij}^k + \sum_{s \in \widehat{N}_d^k} \delta \times Q_{k,s} - \sum_{(i',j') \in \mathcal{A}_t} \lambda_{i'j'} - \sum_{s \in N} B_s \sum_{(i',j') \in \mathcal{A}_{a'}^s} \rho_{i'j'}. \end{aligned} \tag{10}$$

In (10), C_{ij}^k represents the Lagrangian cost of arc $(i,j) \in \mathcal{A}^k$ of train k. Depending on the arc type, different penalty terms have to be considered in C_{ij}^k, as shown in (10). In particular, we distinguish:

1. Departure arcs after a stop, i.e., $(i,j) \in \mathcal{A}_d^s$, $s \in \widehat{N}_d^k$. In this case, we have to consider the passenger departure and arrival delays, a reward (if the arc is selected we do not have to account for abandoned

passengers), and the penalties $\lambda_{i^{'}j^{'}}$ for all violations of incompatibility-arc constraints:

$$C_{ij}^{k} = q_{k,s}^{on} \times \left(\theta(i) - \widehat{T}_{k,s}^{d}\right) + q_{k,s}^{off} \times \left(\theta(i) - \xi_s - \widehat{T}_{k,s}^{a}\right) - \delta \times Q_{k,s} + \sum_{(i^{'},j^{'}) \in \mathcal{A}_t : (i,j) \in \Psi(i^{'},j^{'})} \lambda_{i^{'}j^{'}} \quad (11)$$

In particular, the last term considers all constraints (6) (defined by any arc $(i^{'}, j^{'}) \in \mathcal{A}_t$) in which traveling arc (i, j) appears, and sums the Lagrangian multipliers of each of these constraints.

2. Travel arcs for which the train does not stop at the departure station, i.e., $(i, j) \in \mathcal{A}_t \backslash \mathcal{A}_d^s$, $s \in \widehat{N}_d^k$. In this case, since the train does not stop at the station, we have only to consider the penalties $\lambda_{i^{'}j^{'}}$ for all violations of incompatibility-arc constraints:

$$C_{ij}^{k} = \sum_{(i^{'},j^{'}) \in \mathcal{A}_t : (i,j) \in \Psi(i^{'},j^{'})} \lambda_{i^{'}j^{'}}. \quad (12)$$

3. Waiting arcs (of the first type) corresponding to a stop longer than the minimum dwelling time, i.e., $(i, j) \in \mathcal{A}_{a^{'}}^s$, $s \in \widehat{N}_d^k$. In this case, we have to consider the passenger arrival delays (to complete the computation together with the departure arcs $(i, j) \in \mathcal{A}_d^s$), and the penalties $\rho_{i^{'}j^{'}}$ for all violations of station capacity constraints:

$$C_{ij}^{k} = -q_{k,s}^{off} + \sum_{(i^{'},j^{'}) \in \mathcal{A}_{a^{'}}^{s} : (i,j) \in \varphi(i^{'},j^{'})} \rho_{i^{'}j^{'}}, \quad (13)$$

In particular, the last term considers all constraints (7) (defined by any arc $(i^{'}, j^{'}) \in \mathcal{A}_{a^{'}}^s$) in which waiting arc (i, j) appears, and sums the Lagrangian multipliers of each of these constraints.

4. Other waiting arcs (of the second type) corresponding to a minimum dwelling time stop, i.e., $(i, j) \in \mathcal{A}_a^s \backslash \mathcal{A}_{a^{'}}^s$, $s \in \widehat{N}_d^k$. In this case, we only have to take into account the penalties $\rho_{i^{'}j^{'}}$ for all violations of station capacity constraints:

$$C_{ij}^{k} = \sum_{(i^{'},j^{'}) \in \mathcal{A}_{a^{'}}^{s} : (i,j) \in \varphi(i^{'},j^{'})} \rho_{i^{'}j^{'}}. \quad (14)$$

Note that, although the waiting arcs of the second type do not define station capacity constraints, they can appear in station capacity constraints defined by waiting arcs of the first type, hence, the Lagrangian multiplier must be associated with their cost.

Obviously, the relaxed problem after relaxing constraints (6) and (7) is easier to solve than the original problem. However, by further observing the relaxed model, we can find that, if constraints (2)-(5) are removed, the model can be transformed into a series of shortest path problems, one for each train. Fortunately, constraints (2)-(5) can be easily removed from the model by modifying the space-time network $\mathcal{G}$, as follows:

Constraints (2): by analyzing each train $k \in K$ and each station $s \in \widehat{N}_d^k$ where the train was planned to stop, we mark the outgoing arcs associated with the departure nodes having departure time earlier than the departure time $\widehat{T}_{k,s}^d$ in the planned timetable as infeasible arcs. In the process of searching feasible solutions, it is not allowed to select these infeasible arcs.

Constraints (3): based on the route information of not-directly-affected train $k \in K_0$, we identify all possible routes that train k can use and mark them as feasible routes. In the solving process, train k can only choose arcs from the feasible routes.

Constraints (4): on the basis of identifying the feasible routes of not-directly-affected trains, we further analyze set $\widehat{N}_d^k$ and mark the outgoing arcs associated with the departure nodes from the corresponding station s in set $\widehat{N}_d^k$ as feasible ones, while the outgoing arcs associated with the passing nodes through station s are labeled as infeasible ones, to ensure that constraints (4) are not violated.

Constraints (5): after obtaining disruption information, it is easy to derive the relevant disruption arcs in the constructed space-time network $\mathcal{G}$, which are marked as infeasible ones for all trains to ensure that constraints (5) are satisfied.

By the aforementioned processing, we obtain the feasible space-time arcs of the considered trains, which respect constraints (2)-(5). Therefore, we can directly remove constraints (2)-(5) from the model since they are enforced by graph construction. Furthermore, we observe that, for each train $k \in K$, the subproblem corresponds to finding a shortest path from σ^k to τ^k with arc costs given by C_{ij}^k for each arc $(i,j) \in \mathcal{A}^k$. By solving the $|K|$ subproblems, we obtain a lower bound for the TRP, since obviously (some of) the relaxed constraints can be violated. In order to get a better lower bound, we apply a subgradient optimization procedure (see e.g., Caprara et al. (2002), Meng and Zhou (2014), Zhang et al. (2019)) to iteratively improve the Lagrangian multipliers λ and ρ until a termination condition is reached. In particular, multipliers are updated at iteration $n+1$ from the previous iteration n according to the following expressions:

$$\lambda_{i'j'}^{n+1} = \max\Big\{\lambda_{i'j'}^{n} + \alpha^n \Big[\sum_{k\in K} \sum_{(i,j)\in\Psi(i',j')} y_{ij}^k - 1\Big], 0\Big\} \tag{15}$$

$$\rho_{i'j'}^{n+1} = \max\Big\{\rho_{i'j'}^{n} + \alpha^n \Big[\sum_{k\in K} \sum_{(i,j)\in\varphi(i',j')} y_{ij}^k - B_s\Big], 0\Big\} \tag{16}$$

where α^n is the step parameter that is updated by $\alpha^n = \Omega/(n+1)$ (with Ω a constant value). We consider three termination conditions, and stop as soon as one of them is satisfied: (*i*) the optimality gap between the best upper and lower bounds is smaller than a threshold ε^*; (*ii*) the total number of iterations is greater than the maximum number of iterations $\mathcal{N}^*$; (*iii*) the iteration time exceeds the preset maximum computing time τ^*.

5.2 LR-H algorithm

We use the information of the solutions of model LRM to construct feasible solutions for the original problem during the subgradient optimization procedure, and then return the best feasible solution found. More precisely, the LR-H algorithm, shown in Algorithm 1, consists of six steps, described in the following, that are iteratively executed until a termination condition is satisfied.

Algorithm 1: The LR-H Algorithm

```
Input:
  Space-time network $\mathcal{G} = (\mathcal{V}, \mathcal{A})$.
  The original timetable and passenger demand $Q_{k,s}$ $k \in K$, $s \in N$.
  Arc cost $C_{ij}^k$ in the space-time network for each train $k \in K$, and arc $(i, j) \in \mathcal{A}^k$.
  The disruption information $\mathcal{A}_D$.
Output: The solution of train rescheduling
    • Step 1: Initialization
    Initialize the iteration number $n = 0$;
    Initialize multipliers $\lambda_{ij}$ and $\rho_{ij}$ to 0;
    Initialize the best lower bound $LB^* := -\infty$ and best upper bound $UB^* := M_{init}$;
    Initialize the constraint pool $= \emptyset$.
    • Step 2: Generate the lower bound solution
    For $k \in K$ do
    Compute the shortest path of train $k$ by label setting algorithm with multipliers $\lambda_{ij}^n$ and $\rho_{ij}^n$
    End For
    Compute lower bound $LB^n$ by equation (10) and update $LB^* = \max\{LB^*, LB^n\}$
    • Step 3: Find a feasible upper bound solution
    Find feasible solution based on heuristic Algorithm 2 with multipliers $\lambda_{ij}^n$ and $\rho_{ij}^n$;
    Calculate upper bound $UB^n$ by equation (8) and update $UB^* = \min\{UB^*, UB^n\}$
    • Step 4: Calculate relative gap
    Calculate the relative gap $\varepsilon = (UB^* - LB^*)/UB^*$
    • Step 5: Update constraint pool and Lagrangian multipliers
    Update constraint pool by using a dynamic constraint-generation technique;
    Update $\lambda_{i'j'}^{n+1}$ and $\rho_{i'j'}^{n+1}$ based on the updated constraint pool and subgradient method by equations (15) and (16)
    $n := n + 1$
    • Step 6: Test termination conditions
    Stop as soon as one of the termination conditions is satisfied; Otherwise, go to Step 2.
```

In Step 1, we initialize all elements used by LR-H. In particular, to efficiently handle the relaxed constraints (6) and (7), we employ a *constraint pool* that is dynamically updated. Indeed, the sets of constraints (6) and (7) are too large to be handled explicitly within the short computing time required by the application. Since most of the constraints are inactive for a feasible solution, we apply a dynamic constraint-generation technique. Specifically, in Step 1, the constraint pool is initialized as an empty set, and, at each iteration, it is updated by identifying and adding the constraints (6) and (7) violated by the current relaxed solution (lower bound) to the pool. In addition, we set the initial upper bound value to M_{init}, which is the objective value of a feasible solution obtained by assuming that the trains affected by the disruption in the operation process stop and wait in the safe area (including station and track), until the disruption is over.

In Step 2, we compute a lower bound by solving model LRM through a label setting algorithm: it computes the shortest path for each train $k \in K$ by considering Lagrangian costs C_{ij}^k for each arc $(i, j) \in \mathcal{A}^k$. Note that, in order to speed up the solution process, we first perform topological sorting on the constructed space-time network $\mathcal{G}$ to avoid traversing some invalid nodes when executing the shortest path algorithm. This topological sorting only needs to be carried out once, since the topological structure of the space-time network is not changed during the iterations of the algorithm.

In Step 3, we compute an upper bound (feasible heuristic solution) based on the relaxed solution obtained at Step 2. In particular, we construct two feasible solutions by considering the trains ordered according to two alternative train sequences, $\mathcal{S}^1$ and $\mathcal{S}^2$, and select the better one. In the first sequence $\mathcal{S}^1$, the involved trains are ordered in the increasing order of the Lagrangian cost of the corresponding shortest path computed in Step 2, while in the second one $\mathcal{S}^2$, the involved trains are ordered randomly. Given sequence $\mathcal{S}^1$ or $\mathcal{S}^2$, the involved trains are rescheduled one after another. For each train, we compute a path in the space-time network by a label setting algorithm, which is output as the rescheduled timetable of the train. Every time a train is rescheduled, all incompatible arcs due to constraints (6) and (7) are forbidden in the space-time network for the following trains. If no feasible path is found for a train, we just discard the train sequence. If, for both train sequences, no feasible solution can be found, the LR-H algorithm will continue iteratively with the following steps. We observe that the computation of the feasible solution with train sequence $\mathcal{S}^1$ is independent with the computation with train sequence $\mathcal{S}^2$, so parallel computing technology is used in this part of Step 3 to speed up the solution process. The procedure for computing a feasible heuristic solution is reported in Algorithm 2. In Step 3, we finally update the best upper bound, if a better solution has been found.

Algorithm 2: A heuristic procedure to construct a feasible solution

```
Input:
1: Two space-time network $\mathcal{G}_1 = \mathcal{G}_2 = (\mathcal{V}, \mathcal{A})$ for train sequences $\mathcal{S}^1$ and $\mathcal{S}^2$, respectively.
2: Lagrangian costs of all involved trains obtained in Step 2 of Algorithm 1.
Output: A feasible upper bound
  • Step 1: Generate the train sequences
      Sequence $\mathcal{S}^1$: Sort the trains by increasing order of their Lagrangian costs;
      Sequence $\mathcal{S}^2$: Randomly order the trains;
  • Step 2: Generate a feasible solution by rescheduling trains sequentially
  Parallel calculation for train sequences $\mathcal{S}^1$ and $\mathcal{S}^2$
  For $\mathcal{S} \in \{\mathcal{S}^1, \mathcal{S}^2\}$ do
      Initialize $\ell = 0$
      while $\ell <\mid \mathcal{S} \mid$
          Compute the shortest path of train $k = \mathcal{S}(\ell)$ on the updated space-time network by label setting algorithm
          if no feasible solution is found for train $k$
              break;
          else
              Fix the path of train $k$, and update the space-time network by forbidding the arcs that conflict with the path of train $k$, according to constraints (6) and (7)
          end if
          $\ell := \ell + 1$
      end while
  End For
  Return The better feasible solution generated with train sequences $\mathcal{S}^1$ and $\mathcal{S}^2$
```

In Step 4, the relative gap is computed based on the newly obtained lower and upper bounds. In Step 5, we check whether the solution of model LRM violates any relaxed constraints (6) and (7), and update the constraint pool accordingly. Afterwards, the Lagrangian multipliers are also updated based on the constraint violations. Finally, Step 6 is used to determine whether at least one of the termination conditions is satisfied and the algorithm can be stopped.

6 Computational results

We tested the performance of LR-H for several disruption scenarios based on two different railway networks. The first one (see Section 6.1) is a small artificial network on which the LR-H algorithm is compared with the commercial solver GUROBI 10.0.1 applied to solve model (9). The second network (see Section 6.2) is part of the Chinese railway network: for this large-scale instance, only the LR-H method can be used to obtain the results in short computing times. We performed all computational experiments on a computer with an Intel(R) Core(TM) i9-13900K@3.00GHz CPU and 128GB RAM, using Microsoft Windows 11(64 bit) OS. All methods were implemented in Visual C++ 2022.

6.1 Experiments on a small artificial network

We carried out a series of experiments based on the railway network shown in Figure 1. The distances and running times between consecutive stations are reported in Figure 9: in particular, we assume that the distance and running time between consecutive stations are, respectively, 80 km and 20 min. In addition, we assume that station 3 and connected sections ((2,3), (3,6), (3,4)) are disrupted.

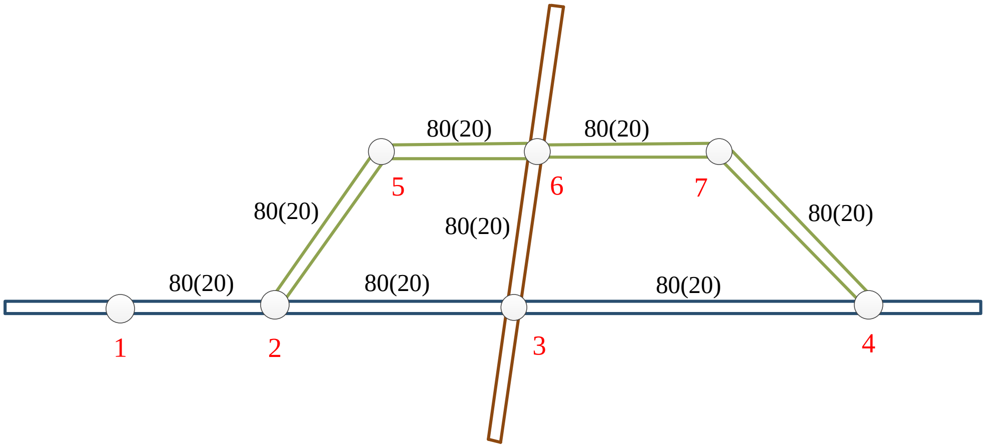


Figure 9: Map of the small artificial network.

In Section 6.1.1, we analyze the impact that different values for the penalty parameter δ, weighting the number of abandoned passengers in the objective function, have on the results. In Section 6.1.2, we show the results of experiments with different disruption scenarios.

6.1.1 Impact of different penalty parameter δ values

We first generated a feasible train timetable, as shown in Figure 10, where a solid line represents a train running on the current line, a dashed line represents a train running on other lines, and the grey area represents the disruption. The figure is divided into three parts, each one corresponding to a possible train path in the network: on top we report the time-space diagram for trains visiting stations 1-2-3-4, in the middle for trains visiting stations 1-2-3-6-7-4, and at the bottom for trains visiting stations 1-2-5-6-7-4. In total, we consider 27 trains running from station 1 to station 4. In this experiment, we set the disruption start and end times to 80 and 110, respectively. In this case, five trains (shown in red in Figure 10) are directly affected by the disruption, and their original timetable is reported in Table 5, where “Arr.” and “Dep.” represent arrival and departure times, respectively.

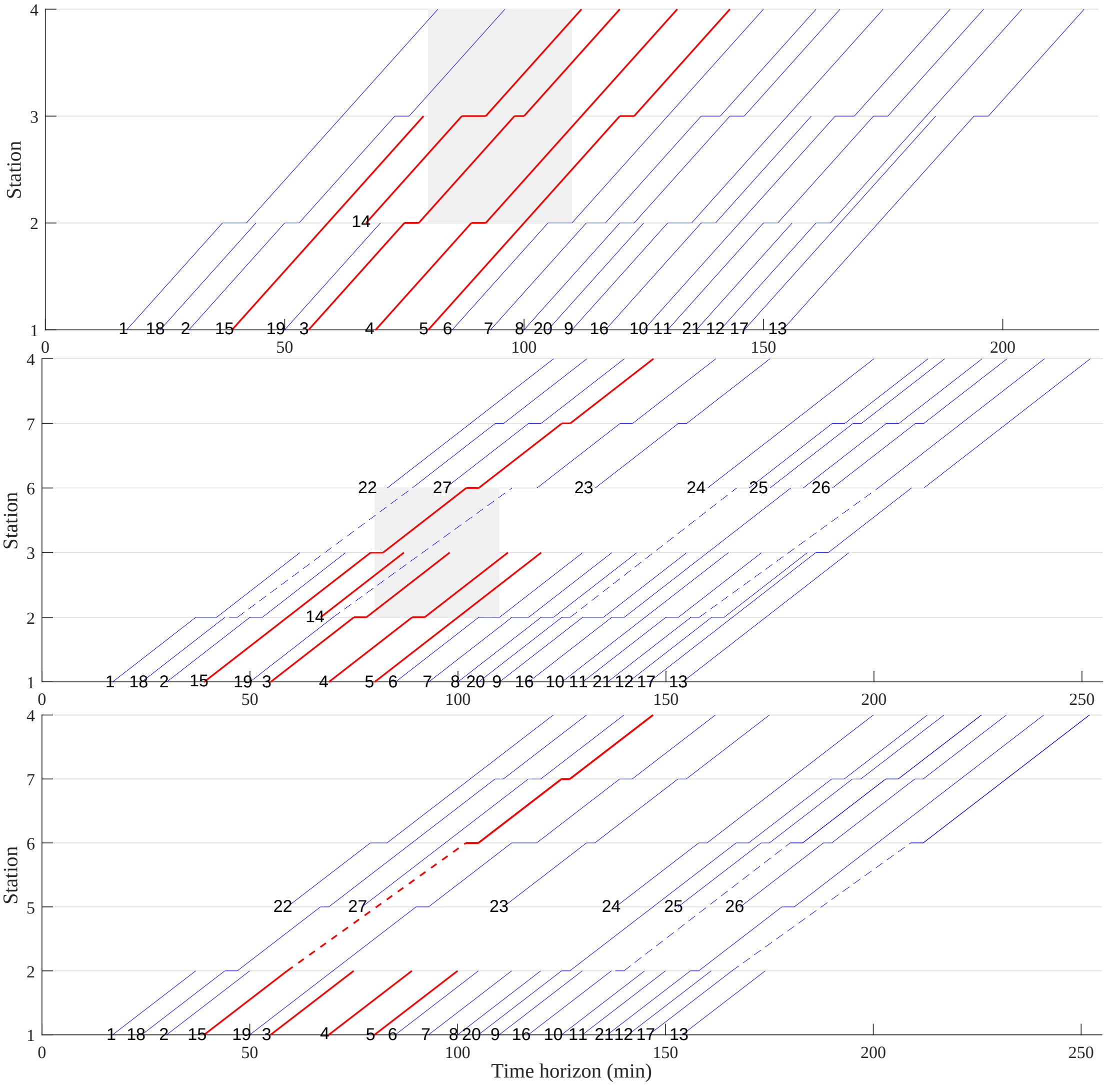


Figure 10: The original train timetable and disruption in the small artificial network.

Table 5: The original timetable of the directly-affected trains.

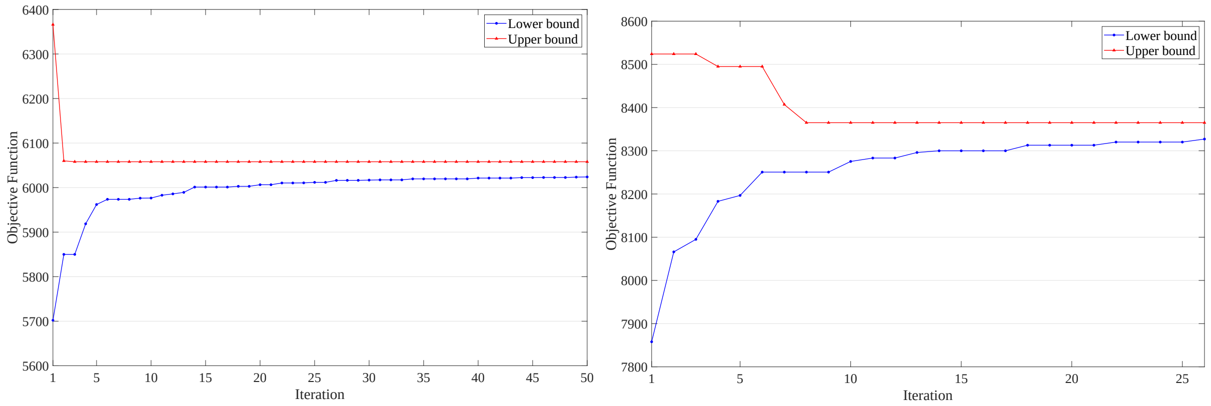
	Train 3	Train 4	Train 5	Train 14	Train 15
Route	$1,2,3,4$	$1,2,3,4$	$1,2,3,4$	$2,3,4$	$1,2,3,6,7,4$
Arr.	$-,75,98,120$	$-,89,112,132$	$-,100,120,143$	$-,87,112$	$-,59,79,102,125,147$
Dep.	$55,78,100,-$	$69,92,112,-$	$80,100,123,-$	$67,92,-$	$39,59,82,105,127,-$

As explained in Section 2, we assume that the routes of the not-directly-affected trains remain unchanged, but the arrival, departure, and dwelling times of these trains can be adjusted. The minimum departure and arrival headways are both set to 2 minutes, and the maximum capacity of station is set to 3. The number of passengers getting on and off each train at each station is randomly generated between 10 and 50. In the LR-H method, the constant Ω, used for updating the step parameter α^n, is set to 20. The termination condition for LR-H for the small artificial network is given by the following parameters: ε^*=0.5%, $\mathcal{N}^* = 50$ and $\tau^* = 10s$ in the termination conditions.

We tested five different δ values. The results obtained by solving the model with GUROBI and by LR-H are reported in Table 6. In addition, Figure 11 plots the convergence of best upper bounds and lower bounds during the iterations for two different δ values when applying the LR-H method.

Table 6: Comparison between GUROBI and the LR-H method for the small railway network.

δ value	GUROBI		LR-H						
	F^*	CPU time	LB	UB	Iter.	Cons.	CPU time	GAP*	GAP
$\delta = 1$	5024	45.78s	5000.42	5024	40	43	0.78s	0.00%	0.47%
$\delta = 10$	6058	49.00s	6023.92	6058	50	52	0.92s	0.00%	0.56%
$\delta = 100$	8365	47.14s	8327.11	8365	26	59	0.48s	0.00%	0.45%
$\delta = 1000$	8365	46.89s	8327.11	8365	26	59	0.49s	0.00%	0.45%
$\delta = 10000$	8365	47.09s	8327.11	8365	26	59	0.48s	0.00%	0.45%



(a) $\delta = 10$

(b) $\delta = 100$

Figure 11: Best upper bounds and lower bounds during the iterations of LR-H with different δ values.

In Table 6, we report in the first column the considered δ value. Then, for the solution obtained by GUROBI, we list the optimal objective value (F^*) and the corresponding computing time (CPU time). For the solution obtained by the LR-H method, we report the best lower bound value LB, the best solution value UB, the total number "Iter." of iterations executed by LR-H, the number "Cons." of constraints in the constraint pool when the algorithm stops, and the computing time (CPU time). Finally, we show two gaps GAP* and GAP that represent, respectively, the relative gap between the best upper bound (UB) and the optimal solution (F^*), and between the best upper bound (UB) and the best lower bound (LB). They are defined as:

$$\text{GAP}^* = \frac{UB - F^*}{UB} \times 100,$$

$$\text{GAP} = \frac{UB - LB}{UB} \times 100.$$

As can be seen from Table 6, the LR-H method can find the optimal solution for all cases in shorter computing time than that of GUROBI. Although LR-H is not able to prove optimality of these solutions, the optimality gap between upper and lower bounds is always below 0.6%. Besides, for penalty values of δ larger or equal to 100, the optimal solution is always the same. As shown in Figure 11 for $\delta = 10$ and $\delta = 100$, in the first iterations the improvement of lower and upper bounds is more significant, while later the curves become smoother, and converge to the best solution value.

In order to more clearly describe the rescheduling strategy determined by LR-H and by GUROBI (that is the same optimal solution for all cases shown in Table 6), Table 7 reports the information of all the adjusted timetables. In particular, we show the train route (that can differ from the one shown in Table 5 due to the disruption), and the arrival and departure times at all visited stations. It is worth noting that, in the above experiments, the timetable of the not-directly-affected trains is not modified in the optimal solution. The reason for this phenomenon is that, in this small case, the total number of trains operating on the railway network is relatively small. Therefore, when five trains are affected by the disruption, there is still enough capacity in the railway network to reschedule the directly-affected trains without modifying the planned timetable of the not-directly-affected trains.

Table 7: The rescheduled timetable of the directly-affected trains.

δ		Train 3	Train 4	Train 5	Train 14	Train 15
	Route	$1, 2, 3, 4$	$1, 2, 5, 6, 7, 4$	$1, 2, 5, 6, 7, 4$	$2, 3, 4$	$1, 2, 3, 4$
1	Arr.	$-, 75, 98, 134$	$-, 89, 112, 135, 155, 177$	$-, 100, 120, 140, 160, 180$	$-, 87, 132$	$-, 59, 79, 147$
	Dep.	$55, 78, 114, -$	$69, 92, 115, 135, 157, -$	$80, 100, 120, 140, 160, -$	$67, 112, -$	$39, 59, 127, -$
	Route	$1, 2, 3, 4$	$1, 2, 5, 6, 7, 4$	$1, 2, 3, 4$	$2, 3, 4$	$1, 2, 3, 4$
10	Arr.	$-, 75, 98, 134$	$-, 89, 112, 135, 157, 177$	$-, 100, 132, 154$	$-, 87, 132$	$-, 59, 79, 147$
	Dep.	$55, 78, 114, -$	$69, 92, 115, 137, 157, -$	$80, 112, 134, -$	$67, 112, -$	$39, 59, 127, -$
	Route	$1, 2, 3, 4$	$1, 2, 5, 6, 7, 4$	$1, 2, 3, 4$	$2, 3, 4$	$1, 2, 3, 6, 7, 4$
100	Arr.	$-, 75, 98, 134$	$-, 89, 112, 137, 157, 179$	$-, 100, 132, 154$	$-, 87, 132$	$-, 59, 79, 132, 155, 177$
	Dep.	$55, 78, 114, -$	$69, 92, 117, 137, 159, -$	$80, 112, 134, -$	$67, 112, -$	$39, 59, 112, 135, 157, -$
	Route	$1, 2, 3, 4$	$1, 2, 5, 6, 7, 4$	$1, 2, 3, 4$	$2, 3, 4$	$1, 2, 3, 6, 7, 4$
1000	Arr.	$-, 75, 98, 134$	$-, 89, 112, 136, 159, 179$	$-, 100, 132, 154$	$-, 87, 132$	$-, 59, 79, 132, 155, 177$
	Dep.	$55, 78, 114, -$	$69, 92, 116, 139, 159, -$	$80, 112, 134, -$	$67, 112, -$	$39, 59, 112, 135, 157, -$
	Route	$1, 2, 3, 4$	$1, 2, 5, 6, 7, 4$	$1, 2, 3, 4$	$2, 3, 4$	$1, 2, 3, 6, 7, 4$
10000	Arr.	$-, 75, 98, 134$	$-, 89, 112, 136, 159, 179$	$-, 100, 132, 154$	$-, 87, 132$	$-, 59, 79, 132, 155, 177$
	Dep.	$55, 78, 114, -$	$69, 92, 116, 139, 159, -$	$80, 112, 134, -$	$67, 112, -$	$39, 59, 112, 135, 157, -$

As shown in Table 7, the routes of trains 3 and 14 cannot be changed, and they have to wait on a track, since they have already entered the disrupted area when the disruption starts (see Figure 10). On the contrary, for the other trains, the route can be modified. When the penalty δ for abandoning passengers is relatively small, trains will more likely choose other routes to reduce passenger delays. In particular, when δ is 1 or 10, train 15 skips stations 6 and 7, so that it can run again as in the original timetable when it arrives at station 4. When δ is very large, trains tend to follow the original routes in order to avoid abandoning passengers. Specifically, when $\delta \geq 100$, the penalization caused by abandoning passengers will be too large, so we have $\sum_{(i,j)\in\mathcal{A}_d^s} y_{ij}^k = 1$ in an optimal solution, i.e., the first term of the objective function becomes 0. Therefore, under this condition, the results correspond only to passenger delays. For example, when $\delta \geq 100$, neither train 5 nor train 15 change the original stop plan. It is also worth noting that, since train 4 has no stop planned at station 3 (see Table 5), in order to reduce passenger dissatisfaction caused by long waiting times at station 2, route 1-2-5-6-7-4 is chosen for the train (i.e., it skips station 3) to ensure the departure from station 2 on time.

6.1.2 Numerical experiments with different disruption scenarios

In this section, we consider $\delta = 100$, and perform experiments under the different disruption scenarios that are shown in Table 8, where we report, for each scenario ω, the start and end times of the disruption and its duration. For all scenarios, we consider the original timetable of Figure 10.

Table 8: Information on the considered disruption scenarios.

Scenario	$\omega1$	$\omega2$	$\omega3$	$\omega4$	$\omega5$	$\omega6$	$\omega7$	$\omega8$	$\omega9$	$\omega10$
Start time	100	100	80	80	60	60	60	50	50	50
End time	130	150	90	130	80	100	120	70	90	110
Duration	30	50	10	50	20	40	60	20	40	60

Table 9 lists the computational results obtained by GUROBI and LR-H under these ten disruption scenarios. All columns have the same meaning as in Table 6, and we also report, in the last row, the averages of computing times for GUROBI and LR-H, and the average percentage gaps. As can be seen from Table 9, although the LR-H method cannot find the optimal solution for all scenarios, it achieves an average gap of 1.01% from the optimal solution value within very short computing time (0.76 sec on average), while GUROBI requires about 52 sec on average. In addition, the gap between the upper and lower bounds is on average 1.89%, showing that the Lagrangian relaxation provides a lower bound of good quality. We observe that the longer the disruption duration is, the more difficult it becomes for LR-H to find the optimal solution: indeed, the larger gaps from the optimal solutions are obtained for scenarios $\omega4$, $\omega7$, $\omega10$, that have long disruption duration. This phenomenon is also in line with reality, because a longer disruption affects more trains, and, consequently, more adjustments are needed.

Table 9: Computational results with different disruption scenarios.

Scenario	GUROBI		LR-H						
	F^*	CPU time	LB	UB	Iter.	Cons.	CPU time	GAP*	GAP
$\omega 1$	3860	38.63s	3841.29	3860	34	31	0.57s	0.00%	0.48%
$\omega 2$	11863	40.03s	11673.60	11909	50	56	0.79s	0.39%	1.98%
$\omega 3$	1451	49.07s	1451	1451	1	0	0.02s	0.00%	0.00%
$\omega 4$	17674	46.61s	17606.10	18284	50	64	0.90s	3.34%	3.71%
$\omega 5$	2017	56.88s	2007	2017	6	24	0.19s	0.00%	0.50%
$\omega 6$	11697	56.87s	11599	11753	50	49	0.94s	0.48%	1.31%
$\omega 7$	22002	56.70s	21911.70	22804	50	65	1.33s	3.52%	3.91%
$\omega 8$	5047	57.10s	4930.11	5047	50	26	0.97s	0.00%	2.32%
$\omega 9$	16241	56.35s	15876.2	16269	50	49	0.96s	0.17%	2.41%
$\omega 10$	24627	57.70s	24604.30	25180	50	59	0.96s	2.17%	2.29%
Avg.		51.59s					0.76s	1.01%	1.89%

Table 10 shows the rescheduling strategies selected in the best solution found, respectively, by GUROBI and LR-H under the ten disruption scenarios. In the table, "DT" and "AT" represent, respectively, the directly-affected trains and those for which the timetable is adjusted in the rescheduling process. In addition, "Re-T" (retiming) means that the train does not change its original route, but the arrival time and/or departure time in the original timetable are modified, while "Re-R" (rerouting) means that the train follows an alternative route.

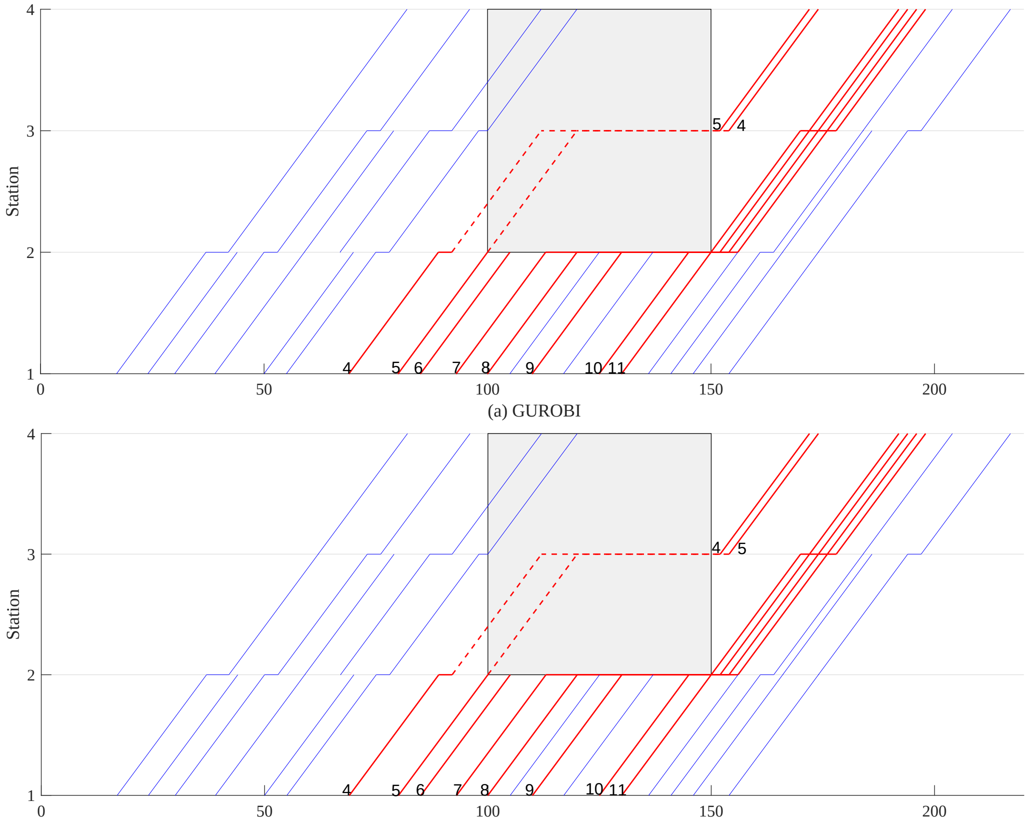
Table 10: Rescheduling strategies selected by GUROBI and LR-H under different disruption scenarios.

Scenario	DT	GUROBI			LR		
		AT	Re-T	Re-R	AT	Re-T	Re-R
$\omega 1$	4,5, 6,7,8	4,5,6, 7,8,9	4,5, 7,8,9	6	4,5,6 7,8,9	4,5, 7,8,9	6
$\omega 2$	4,5,6,7, 8,9,10,16	4,5,6,7,8, 9,10,11,16*	4,5,7, 8,10,11	6,9, 16($f^*_{16} = 0$)	4,5,6,7,8, 9,10,11,16*	4,5,7, 8,10,11	6,9, 16($f^*_{16} = 0$)
$\omega 3$	15	15	15	–	15	15	–
$\omega 4$	3,4,5,6, 7,8,14,15	3,4,5,6, 7,8,9,14,15	3,5,7, 8,9,14,15	4,6	3,4,5,6, 7,8,9,14,15	3,5,7, 8,9,14,15	4,6
$\omega 5$	1,2, 3,14	1,2, 3,14	1,2, 3,14	–	1,2, 3,14	1,2, 3,14	–
$\omega 6$	1,2,3, 4,14,15	1,2,3, 4,5,14,15	1,2,3, 4,5,14,15	–	1,2,3, 4,5,14,15	1,2,3, 4,5,14,15	–
$\omega 7$	1,2,3,4,5, 6,7,14,15	1,2,3,4,5, 6,7,8,14,15	1,2,3, 5,7,8,15	4,6,14	1,2,3,4,5, 6,7,8,14,15	1,2,3, 5,7,8,15	4,6,14
$\omega 8$	1,2, 14,15	1,2, 14,15	1,2, 14,15	–	1,2, 14,15	1,2, 14,15	–
$\omega 9$	1,2, 3,14,15	1,2,3, 4,14,15	1,2, 3,4,14	15	1,2,3 4,14,15	1,2, 3,4,14	15
$\omega 10$	1,2,3, 4,5,14,15	1,2,3,4, 5,6,14,15	1,3, 5,6	2,4, 14,15	1,2,3,4, 5,6,14,15	1,3, 5,6	2,4, 14,15

From Table 10, we can see that, in the 10 scenarios, the rescheduling strategies selected in the best solution found by the two methods are the same: compared with GUROBI, LR-H can find the correct rescheduling strategy faster, even if, in some cases (i.e., scenarios 2, 4, 6, 7, 9 and 10), there are small deviations in the adjustment of arrival or departure time, resulting in an objective value slightly larger than the optimal one.

We observe that, for most trains, the retiming rescheduling strategy is selected. However, it is also worth noting that, in scenario $\omega 2$, in which the disruption starts at time instant 100 and ends at 150, if train 16 is scheduled according to the original timetable, it will inevitably be affected by the disruption (see Figure 10). However, because train 16 has no stop planned at station 3, it can avoid being delayed by choosing the rerouting strategy (i.e., the delay f^*_{16} of train 16 is 0). This also shows that, when the railway network is subject to a large-scale disruption, it is important to introduce the rerouting strategy.

Figure 12 plots the rescheduled time-space diagrams of the two solving methods (GUROBI and LR-H) for the line including stations 1, 2, 3 and 4 in scenario $\omega 2$.



(a) GUROBI

(b) LR

Figure 12: The rescheduled timetables in scenario $\omega 2$.

In this case, the total passenger delay obtained by GUROBI is 11863, given by the sum of the delays of trains 4, 5, 6, 7, 8, 9, 10, 11 and 16, which are respectively 1092, 1421, 440, 3986, 3333, 880, 648, 63 and 0. The total passenger delay obtained by LR-H is 11909: the same trains are delayed and by the same amount except for trains 4 and 5 whose delay is, respectively, 1040 and 1519. As can be seen in Figure 12, the departure sequence of trains 4 and 5 obtained by LR-H is different from the one computed by GUROBI, that coincides with the optimal plan. This results in a slightly larger total passenger delay for LR-H. However, LR-H requires much shorter computing times, and this is crucial, especially for large-scale instances, for which GUROBI is no longer able to solve the model within acceptable computing times.

6.2 Large-scale experiment

6.2.1 Experiment description

In this section, we report the results obtained by the LR-H method on a real-world instance of the Chinese railway network with five double-track lines, marked with different colors in Figure 13. The tested railway network consists of 47 stations and 50 sections with a total length of 1935 km, of which lines 1, 2, 3, 4 and 5 are, respectively, 247 km, 606 km, 442 km, 340 km and 300 km. Figure 13 indicates the names of the stations and the corresponding index in brackets. Besides, the number marked between two consecutive stations is the corresponding distance (in km). For example, the distance of section ‘10 ↔ 11’ on L2 is 101, which means that the distance from station Nanchang West (10) to station Fuzhou (11) is 101 km.

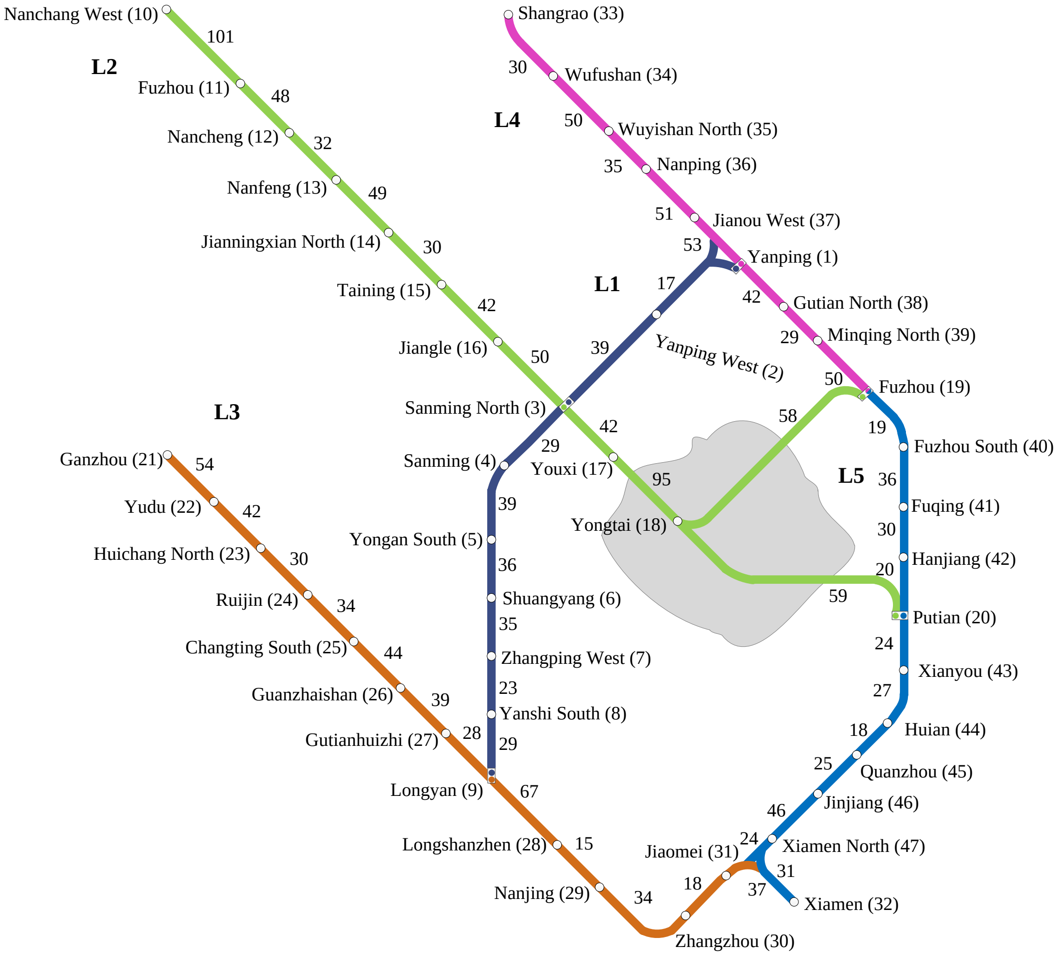


Figure 13: Map of the considered real-world railway network.

We use the actual timetable of March 21, 2020 as the original timetable. The planning time horizon H we consider is [6:00, 24:00] (i.e., [0 min, 1080 min]). During this planning horizon, in the network 350 trains travel, of which 188 trains traverse at least two lines of the network, accounting for 53.7% of the total. This confirms that it is important to consider the rescheduling problem in the whole network.

Based on the actual timetable, we derived the upper and lower limits for the running time along each section, as shown in Table 11. For example, in the first row, $(4, 11)$ means that the shortest and longest allowed running times from Yanping to Yanping West are, respectively, 4 and 11 minutes. In Table 11, we can see that the difference between upper and lower running time limits of many sections, such as $32 \to 31$ and $47 \to 32$, is very large. This leads to a very large number of arcs in the space-time network, which greatly increases the scale of the problem.

In this instance, the minimum departure and arrival headways are both set to 3 minutes, the minimum dwelling time at every station is set to 2 minutes, and the maximum station capacity is set to 6. Because of the protection of commercial data, it was not possible for us to obtain the corresponding passenger demand. Therefore, we randomly generated passenger demands. Specifically, the number of passengers getting on and off each train at each station is randomly generated between 20 and 100, and the total number of passengers is 270636. In the LR-H method, the constant Ω for updating the step parameter α^n is set to 20. Unless otherwise stated, we set $\varepsilon^* = 1.0\%$, $\mathcal{N}^* = 50$ and $\tau^* = 600$s in the following experiments (but we also consider shorter time limits in Section 6.2.3). We consider that the disruption occurs at station Yongtai and connected sections, which is illustrated by the shadow in Figure 13.

In the following, we report the results obtained by LR-H with four experiments, namely, using different values for the penalty parameter δ (Section 6.2.2), considering different disruption scenarios (Section 6.2.3), evaluating the impact of allowing or not rerouting as a rescheduling strategy (Section 6.2.4), and

Table 11: Running times in the actual railway network.

Sec.	Time	Sec.	Time	Sec.	Time	Sec.	Time	Sec.	Time
$1 \to 2$	$(4, 11)$	$9 \to 27$	(10,16)	$19 \to 18$	$(20, 26)$	$28 \to 29$	$(5, 9)$	$38 \to 39$	$(7, 12)$
$1 \to 37$	$(13, 17)$	$9 \to 28$	(24,35)	$19 \to 39$	$(13, 20)$	$29 \to 28$	$(5, 9)$	$39 \to 19$	$(13, 23)$
$1 \to 38$	$(11, 17)$	$10 \to 11$	(33,40)	$19 \to 40$	$(7, 17)$	$29 \to 30$	$(12, 21)$	$39 \to 38$	$(7, 12)$
$2 \to 1$	$(4, 11)$	$11 \to 10$	(33,43)	$20 \to 18$	$(20, 30)$	$30 \to 29$	$(12, 17)$	$40 \to 19$	$(7, 17)$
$2 \to 3$	$(10, 16)$	$11 \to 12$	(16,21)	$20 \to 42$	$(6, 10)$	$30 \to 31$	$(6, 12)$	$40 \to 41$	$(11, 20)$
$3 \to 2$	$(11, 17)$	$12 \to 11$	(16,21)	$20 \to 43$	$(8, 12)$	$31 \to 30$	$(6, 12)$	$41 \to 40$	$(12, 18)$
$3 \to 4$	$(9, 16)$	$12 \to 13$	(10,14)	$21 \to 22$	$(20, 28)$	$31 \to 32$	$(19, 25)$	$41 \to 42$	$(10, 15)$
$3 \to 16$	$(16, 22)$	$13 \to 12$	(11,14)	$22 \to 21$	$(26, 28)$	$31 \to 47$	$(9, 16)$	$42 \to 20$	$(6, 11)$
$3 \to 17$	$(14, 20)$	$13 \to 14$	(16,20)	$22 \to 23$	$(16, 17)$	$32 \to 31$	$(16, 28)$	$42 \to 41$	$(10, 14)$
$4 \to 3$	$(8, 15)$	$14 \to 13$	(16,20)	$23 \to 22$	$(16, 18)$	$32 \to 47$	$(15, 25)$	$43 \to 20$	$(8, 11)$
$4 \to 5$	$(14, 16)$	$14 \to 15$	(10,14)	$23 \to 24$	$(11, 15)$	$33 \to 34$	$(7, 12)$	$43 \to 44$	$(9, 13)$
$5 \to 4$	$(13, 18)$	$15 \to 14$	(10,14)	$24 \to 23$	$(11, 16)$	$34 \to 33$	$(7, 11)$	$44 \to 43$	$(9, 13)$
$5 \to 6$	$(13, 16)$	$15 \to 16$	(14,18)	$24 \to 25$	$(13, 16)$	$34 \to 35$	$(12, 16)$	$44 \to 45$	$(5, 10)$
$6 \to 5$	$(12, 16)$	$16 \to 3$	(16,21)	$25 \to 24$	$(12, 16)$	$35 \to 34$	$(12, 18)$	$45 \to 44$	$(6, 10)$
$6 \to 7$	$(13, 15)$	$16 \to 15$	(14,19)	$25 \to 26$	$(16, 19)$	$35 \to 36$	$(8, 13)$	$45 \to 46$	$(8, 12)$
$7 \to 6$	$(12, 15)$	$17 \to 3$	(14,20)	$26 \to 25$	$(16, 18)$	$36 \to 35$	$(9, 13)$	$46 \to 45$	$(7, 13)$
$7 \to 8$	$(8, 13)$	$17 \to 18$	(31,34)	$26 \to 27$	$(14, 18)$	$36 \to 37$	$(12, 18)$	$46 \to 47$	$(15, 25)$
$8 \to 7$	$(8, 11)$	$18 \to 17$	(32,35)	$27 \to 9$	$(10, 16)$	$37 \to 1$	$(13, 18)$	$47 \to 31$	$(9, 13)$
$8 \to 9$	$(11, 17)$	$18 \to 19$	(20,27)	$27 \to 26$	$(14, 18)$	$37 \to 36$	$(12, 16)$	$47 \to 32$	$(14, 30)$
$9 \to 8$	$(10, 17)$	$18 \to 20$	(20,26)	$28 \to 9$	$(24, 29)$	$38 \to 1$	$(10, 16)$	$47 \to 46$	$(13, 23)$

evaluating the performance of the dynamic constraint-generation technique (Section 6.2.5). Due to the size of the real-world instance, we do not report any comparison with GUROBI.

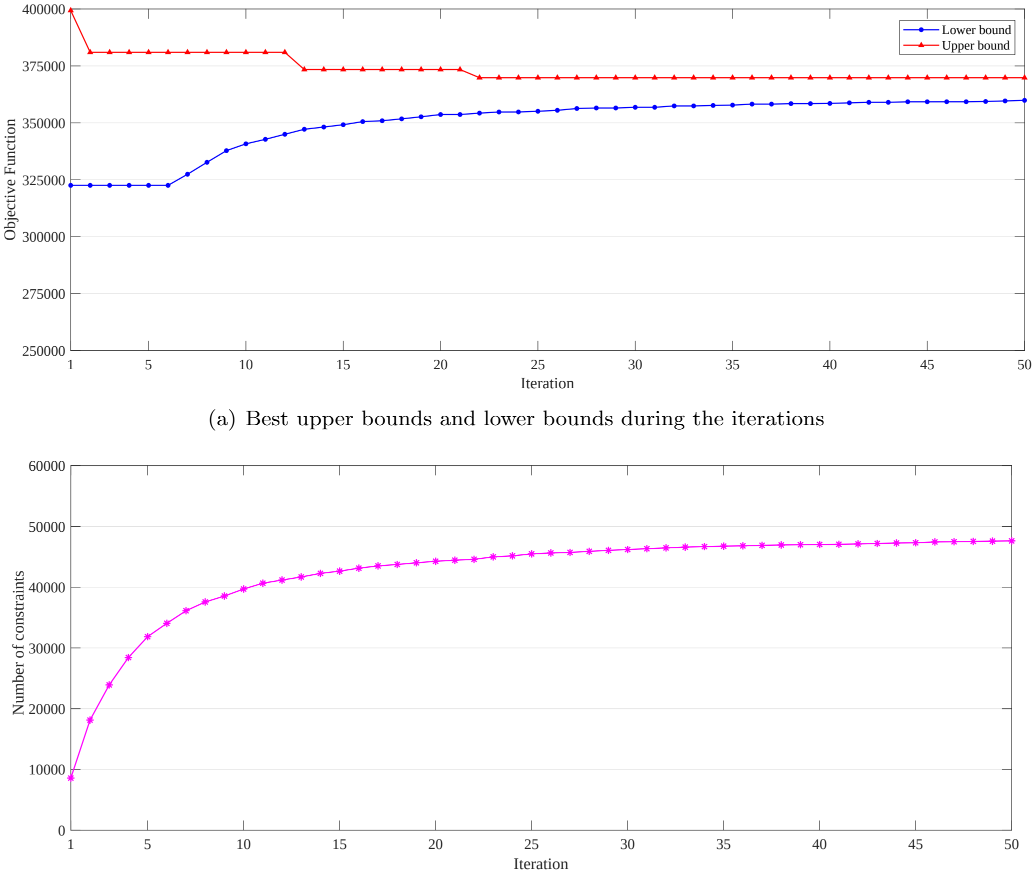
6.2.2 Analysis of different δ values in the actual network

In this experiment, we set the disruption start and end times, respectively, as minutes 420 and 600 of the time horizon. The duration of the disruption is thus 3 hours, and a total of 18 trains are directly affected. Table 12 shows the computational results obtained with different δ values. The first column shows the considered value for δ, and the second column reports the objective value F of the best solution found by LR-H, i.e., $F = \delta f_1 + f_2$, where f_1 represents the number of abandoned passengers, and f_2 represents the total passenger delay. The third and fourth columns report f_1 and f_2, respectively. In addition, in the third column, we show in brackets the value "*Per.*" that represents the percentage of abandoned passengers with respect to the total number of passengers. In the fourth column, we display in brackets the value "*Avg.*" that represents the average delay in minutes for all delayed passengers: it is calculated as the ratio of the sum of terms 2 and 3 in (8) (i.e., f_2 in Table 12) and the number of all involved delayed passengers "$\#DP$". Here, $\#DP$ represents the total number of delayed passengers getting on (i.e., $q^{on}_{k,s}$) and off (i.e., $q^{off}_{k,s}$) delayed trains. Then, in Table 12, we display the number "#AT" of trains whose timetable was adjusted in the solution, and the numbers "#Re-T" and "#Re-R", representing, respectively, how many trains were only retimed and how many were rerouted. Finally, we show the number "Cons." of constraints in the constrain pool when LR-H terminates, the number "Iter." of iterations executed by LR-H, its computing time, and the percentage gap between the best solution found and the best lower bound.

Table 12: Computational results of LR-H with different δ values in the actual network.

δ	F	$f_1(Per.)$	$f_2(\#DP, Avg.)$	#AT	#Re-T	#Re-R	Cons.	Iter.	CPU time	GAP
1	196926	3244(1.20%)	$193682(3909, 49.55)$	31	17	14	37923	22	119s	0.77%
10	220655	2376(0.88%)	$196895(5098, 38.62)$	33	20	13	41146	18	97s	0.70%
100	369825	1222(0.45%)	$247625(8853, 27.97)$	36	25	11	47620	50	299s	2.69%
1000	682172	0(0%)	$682172(11990, 56.90)$	36	27	9	66781	50	454s	3.64%
10000	685895	0(0%)	$685895(12833, 53.45)$	32	23	9	66793	50	381s	4.16%

As can be seen from Table 12, for all cases, the obtained solutions have an optimality gap below 4.5% and are obtained within 500 seconds. We also observe that, for $\delta \leq 100$, the optimality gaps are smaller, and the computing times are shorter; for $\delta \geq 10$, the percentage of abandoned passengers is less than 1%. With the increase of penalty parameter δ, fewer and fewer trains choose the rerouting strategy, and, for large values of δ (e.g., 1000 or 10000), no passengers are abandoned: in this case, trains can still be rerouted, but only if the passengers do not get on or off the trains at the skipped stations. We note that, when $\delta \geq 1000$, the total delay increases. Indeed, to avoid abandoning passengers, trains cannot be rerouted and have to wait longer times. In practice, the railway company can choose the appropriate value of δ, according to the actual needs to obtain a more suitable rescheduled timetable. Based on our results, an appropriate value for δ appears to be 100, as it is a good compromise for delay reduction while limiting the number of abandoned passengers. For $\delta = 100$, we further illustrate in Figure 14(a) the plots of the best upper and lower bounds, and in Figure 14(b) the number of constraints in the constraint pool as a function of the number of iterations.



(a) Best upper bounds and lower bounds during the iterations

(b) The number of constraints during the iterations

Figure 14: Variation of the best upper and lower bounds, and the number of constraints in the constraint pool when $\delta = 100$.

It can be seen from Figure 14 that the change trend of the lower bound is consistent with the change trend of the number of constraints in the constraint pool. In early iterations, more constraints have to be dynamically added to the constraint pool, hence, the number of constraints increases faster, while later it increases slightly because there are less constraints outside the constraint pool. Since the Lagrangian multipliers are updated based on the changes in the constraint pool, the lower bound improvement follows a similar trend. In addition, it must be mentioned that the number of constraints finally added to the constraint pool accounts for only 1.7% of the total number of relaxed constraints. Specifically, the numbers of relaxed constraints (6) and (7) are, respectively, $|\mathcal{A}_t|$ and $|\mathcal{A}_{a'}|$, where $|\mathcal{A}_t|$ and $|\mathcal{A}_{a'}|$ represent, respectively, the total number of traveling arcs and the total number of waiting arcs of the first type. After calculation, the sum of $|\mathcal{A}_t|$ and $|\mathcal{A}_{a'}|$ is up to 2.8 million, while there are only $47,620$ constraints in the constraint pool. Therefore, the number of constraints added to the constraint pool is very small in proportion to the total number of relaxed constraints, indicating that the dynamic constraint-generation technique is crucial for the studied problem.

Figure 15 shows the rescheduled timetable with $\delta = 100$, represented by space-time diagram on line L2. In the figure, trains that are scheduled as in the original timetable are shown with black lines, trains that are rerouted are shown with blue lines, and trains for which retiming is applied are shown with red lines. We observe that, the disruption ends at time 600, but the rescheduling of some trains are much later, which implies that the impact of the disruption may last for a long time. This particularly occurs when rescheduling long-distance trains that more likely affect subsequent trains. Hence, we consider all trains as candidate trains to be rescheduled. In this way, we do not need to fix the end of the rescheduling horizon in advance, which would also be a hard task. In addition, considering a longer rescheduling horizon can allow for finding better solutions.

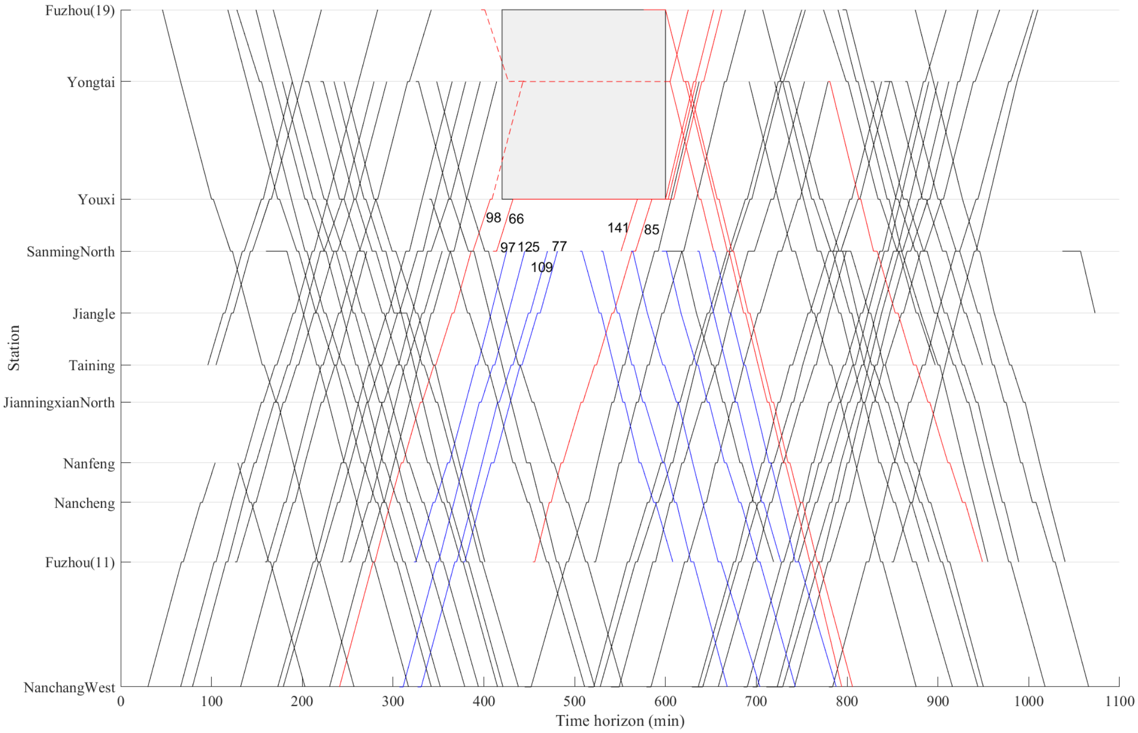


Figure 15: The rescheduled timetable (L2) with penalty parameter $\delta = 100$.

To illustrate the train rescheduling strategy, we select the 8 trains in Figure 15 whose indexes are reported. Firstly, we observe that for trains 98 and 66 there is no choice of rerouting strategy, since the disruption has not yet occurred when they depart from station Sanming North. As for the other trains, the choice of rerouting mainly depends on how long the train has to wait before the disruption is over. In general, if the train has to wait for a long time, then, rerouting is an effective option, so that it can quickly run again as in the original timetable: this is what happens for trains 77, 97, 109 and 125. On the contrary, trains 85 and 141 have to wait for a short time, so the rerouting strategy is not chosen for them. To describe the rescheduling strategy of the 8 trains in detail, we show, in Table 13, the comparison between their original and their rescheduled timetable. In the table, “Ori.” and “Re.” represent, respectively, the original timetable and the rescheduled one. In addition, the first number in the timetable represents the station index, and the two numbers in brackets represent, respectively, the arrival and departure times. Note that only part of the timetable of the train after Sanming North is reported.

As already mentioned, trains 66 and 98 wait until the disruption has been resolved, hence their route is not changed. For train 97 the disruption information has been received when it arrives at station Sanming North, so we can flexibly choose the strategy of waiting or rerouting: as a result, for train 97 the rerouting is applied, and it can travel again as in the original timetable from station 19, although the

Table 13: Comparison of train timetables before and after rescheduling of the 8 selected trains.

Train		Timetable
66	Ori.	...3(410, 414) → 17(432, 434) → 18(468, 470) → 19(495)
	Re.	...3(410, 414) → 17(432, 600) → 18(631, 633) → 19(653)
77	Ori.	...3(481, 485) → 17(503, 505) → 18(536, 536) → 20(555, 555) → 43(563, 563) → 44(572, 572) → 45(578, 580) → 46(589, 589) → 47(606, 611) → 32(638)
	Re.	...3(481, 485) → 4(494, 494) → 5(509, 509) → 6(522, 522) → 7(535, 535) → 8(543, 543) → 9(558, 558) → 28(582, 582) → 29(587, 589) → 30(601, 601) → 31(607, 607) → 47(616, 618) → 32(638)
85	Ori.	...3(563, 567) → 17(585, 587) → 18(621, 623) → 20(644, 644) → 43(653, 655) → 44(666, 666) → 45(673, 675) → 46(684, 684) → 47(700, 703) → 31(714, 714) → 30(722)
	Re.	...3(563, 567) → 17(585, 603) → 18(634, 640) → 20(662, 662) → 43(670, 676) → 44(685, 685) → 45(693, 695) → 46(703, 703) → 47(721, 723) → 31(732, 732) → 30(738)
97	Ori.	...3(425, 427) → 17(446, 448) → 18(482, 482) → 19(503)
	Re.	...3(425, 427) → 2(438, 438) → 1(449, 449) → 38(466, 466) → 39(478, 480) → 19(503)
98	Ori.	...3(386, 388) → 17(407, 409) → 18(443, 443) → 19(463)
	Re.	...3(386, 388) → 17(407, 409) → 18(443, 605) → 19(625)
109	Ori.	...3(470, 474) → 17(488, 488) → 18(521, 521) → 20(541, 558) → 43(567, 567) → 44(577, 577) → 45(584, 586) → 46(595, 595) → 47(611)
	Re.	...3(470, 474) → 2(485, 485) → 1(489, 489) → 38(500, 500) → 39(507, 507) → 19(520, 520) → 40(527, 527) → 41(538, 538) → 42(548, 548) → 20(554, 558) → 43(566, 566) → 44(575, 575) → 45(584, 586) → 46(594, 594) → 47(611)
125	Ori.	...3(445, 449) → 17(464, 464) → 18(497, 497) → 19(518)
	Re.	...3(445, 449) → 2(465, 467) → 1(478, 480) → 38(497, 503) → 39(510, 510) → 19(523)
141	Ori.	3(551) → 17(569, 571) → 18(604, 606) → 19(631)
	Re.	3(551) → 17(569, 609) → 18(640, 642) → 19(662)

passengers who wanted to get on and off the train at station 17 can not be served. A similar choice is done for trains 77, 109 and 125. However, for train 141, because of the passenger demand at stations 17 and 18, and because the waiting time is short, the choice is that it waits at station 17 until the disruption is over. A similar behavior is observed for train 85.

In addition, we can see that different routes are selected for the rerouted trains, based on the station capacity limit and passenger demand. Since there are a large number of trains served by lines 4 and 5, some trains are also rerouted on line 3, if the number of abandoned passengers is low (see e.g., train 77). It should also be pointed out that the not-directly-affected trains are only slightly adjusted. Observing Figure 15, it can be seen that there is an adjusted train (i,e., red train path) running around 800-950 min from Yongtai (18) to Fuzhou (11). By analyzing the original timetable, we find that this train is from Xiamen (32) to Fuzhou (11), which runs on lines $L5$ and $L2$. Because line $L5$ serves some additional rerouted trains, the arrival and departure times of the train are slightly affected. This also shows that it is important to consider a longer time horizon for the problem of large-scale disruption in a railway network.

6.2.3 Numerical experiments with different disruption scenarios in the actual network

In this section, we fix the penalty parameter δ to 100, and perform numerical experiments for 15 different disruption scenarios shown in Table 14. These scenarios are characterized by four different sets of disruption start times (i.e., 60, 240, 420 and 600), which cover the morning, noon and afternoon of the day. Based on the start times, we also set different disruption durations to better analyze the impact of disruption start time and duration on the results.

Table 14: Information on the considered disruption scenarios in the actual network.

Scenario	$\omega1$	$\omega2$	$\omega3$	$\omega4$	$\omega5$	$\omega6$	$\omega7$	$\omega8$	$\omega9$	$\omega10$	$\omega11$	$\omega12$	$\omega13$	$\omega14$	$\omega15$
Start time	60	60	60	240	240	240	240	240	420	420	420	420	420	600	600
End time	240	300	360	360	420	480	540	600	540	600	660	720	780	780	840
Duration	180	240	300	120	180	240	300	360	120	180	240	300	360	180	240

Table 15 displays the computational results obtained by the LR-H algorithm for these 15 disruption scenarios. In the table, all columns have the same meaning as Table 12, and we also show "#DT" that represents the number of the directly-affected trains.

Table 15: Computational results of LR-H with different disruption scenarios.

Scen.	F	$f_1(Per.)$	$f_2(Avg.)$	#DT	#AT	#Re-T	#Re-R	Cons.	Iter.	CPU time	GAP
$\omega1$	269597	1075(0.40%)	162097(21.54)	13	35	26	9	51911	14	194s	0.97%
$\omega2$	386737	1490(0.55%)	237737(24.38)	16	44	33	11	62873	50	472s	2.75%
$\omega3$	479233	2335(0.86%)	245733(24.81)	20	55	37	18	71246	50	520s	1.05%
$\omega4$	267739	383(0.14%)	229439(49.93)	9	20	17	3	41312	9	67s	0.45%
$\omega5$	420662	834(0.31%)	337262(60.70)	13	30	23	7	46365	9	67s	0.56%
$\omega6$	629060	1246(0.46%)	504460(61.94)	19	42	32	10	53333	12	95s	0.60%
$\omega7$	859892	2516(0.93%)	608292(67.63)	23	49	33	16	66669	50	420s	2.37%
$\omega8$	1057778	2680(0.99%)	789778(61.07)	29	55	35	20	75200	50	466s	2.57%
$\omega9$	223516	699(0.26%)	153616(35.32)	12	19	12	7	39625	37	189s	0.98%
$\omega10$	369825	1222(0.45%)	247625(27.97)	18	36	25	11	47620	50	299s	2.69%
$\omega11$	510789	1869(0.69%)	323889(40.16)	20	35	20	15	56097	50	300s	3.26%
$\omega12$	622355	2334(0.86%)	388955(40.37)	27	46	28	18	60306	50	313s	4.81%
$\omega13$	800913	2539(0.94%)	547013(37.06)	30	55	35	20	67898	50	325s	6.42%
$\omega14$	569457	419(0.15%)	527557(52.81)	16	32	27	5	32622	50	178s	2.00%
$\omega15$	855173	1844(0.68%)	670773(69.61)	22	34	21	13	44680	50	303s	2.62%
Avg.										281s	2.27%

The LR-H algorithm obtains near-optimal solutions with an average optimality gap of 2.27% within an average computing time of about 300 seconds, and, in most cases, the optimality gap is smaller than 3.5%. As expected, the longer the disruption duration is, the longer the computing time is needed, and the larger the number of constraints are added to the constraint pool (see, e.g., scenarios $\omega3$, $\omega7$, $\omega8$, $\omega12$, $\omega13$). This is because if the disruption duration becomes longer, more trains are directly affected, and there will be more conflicts when rescheduling involved trains.

In practice, train operators require to obtain an effective rescheduling strategy as quickly as possible. To further analyze the performance of LR-H, we carried out numerical experiments with different values of time limit τ^*. We consider the same 15 scenarios, and set τ^* alternatively to 1 min, 5 min and 10 min, while keeping the other two termination conditions (i.e., $\varepsilon^* = 1.0\%$, $\mathcal{N}^* = 50$) unchanged. Table 16 reports the corresponding computational results, where "$-$" indicates that the LR-H algorithm had already terminated (hence, the values are the same as those obtained with the previous shorter time limit). In the table, F_1 and F_5 represent the objective function value of the best solution obtained within the corresponding time limits of 1 and 5 min, respectively; F is the objective function value of the best solution obtained with a time limit of 10 min (as shown in Table 15). Beside the optimality gap "GAP" in different time limit, we also report the relative gap "F-Gap" that represents the gap between F and F_n, where $n = 1, 5$, i.e.,

$$F\text{-Gap} = \frac{F_n - F}{F_n} \times 100.$$

In the last row, we report the average values of the gaps.

We observe that, when $\tau^* = 1$ min, although the average optimality gap is quite large (i.e., 8.32%), the quality of the obtained feasible solutions is good, i.e., the average value of F-Gap is 1.05%. Moreover, for instances $\omega1$ - $\omega7$ and $\omega11$, the best obtained feasible solutions are as good as when $\tau^* = 10$ min, which indicates that high quality solutions can be found in very short computing times. When $\tau^* = 5$ min, the average value of F-Gap decreases to 0.08%, and meanwhile the average optimality gap significantly reduces to 2.51%. This shows that lower and upper bounds are both effectively improved during the subgradient optimization iterations.

6.2.4 Numerical experiments with different rescheduling strategies

To further highlight the importance of the rerouting strategy for reducing passenger delays, we compare the solutions obtained by employing rerouting with those in which rerouting is not applied. When rerouting is not allowed, trains have to wait at a station or track until the end of the disruption.

Table 16: Computational results of LR-H with different time limit τ^*.

Scenario	τ^*=1 min			τ^*=5 min			τ^*=10 min		
	F_1	GAP	F-Gap	F_5	GAP	F-Gap	F	GAP	F-Gap
ω1	269597	11.05%	0.00%	269597	0.97%	0.00%	–	–	0.00%
ω2	386737	14.09%	0.00%	386737	3.56%	0.00%	386737	2.75%	0.00%
ω3	479233	11.04%	0.00%	479233	1.91%	0.00%	479233	1.05%	0.00%
ω4	267739	2.50%	0.00%	267739	0.45%	0.00%	–	–	0.00%
ω5	420662	2.17%	0.00%	420662	0.56%	0.00%	–	–	0.00%
ω6	629060	3.24%	0.00%	629060	0.60%	0.00%	–	–	0.00%
ω7	859892	8.01%	0.00%	859892	2.65%	0.00%	859892	2.37%	0.00%
ω8	1069910	9.56%	1.13%	1069910	4.03%	1.13%	1057778	2.57%	0.00%
ω9	240106	11.97%	6.91%	223516	0.98%	0.00%	–	–	0.00%
ω10	380960	10.55%	2.92%	369825	2.69%	0.00%	–	–	0.00%
ω11	510789	8.15%	0.00%	510789	3.26%	0.00%	–	–	0.00%
ω12	633565	10.78%	1.77%	622355	4.90%	0.00%	622355	4.81%	0.00%
ω13	811861	11.77%	1.35%	800913	6.50%	0.00%	800913	6.42%	0.00%
ω14	572644	3.71%	0.56%	569457	2.00%	0.00%	–	–	0.00%
ω15	864196	6.26%	1.04%	855173	2.62%	0.00%	–	–	0.00%
Avg.		8.32%	1.05%		2.51%	0.08%		2.27%	0.00%

In these experiments, we consider five scenarios (ω9, ω10, ω11, ω12 and ω13) having disruption start time in the middle of the day (i.e., 420min), and compare the results obtained by LR-H for the two strategies. The results are shown in Tables 17 and 18. In particular, Table 17 reports the results related to the objective function terms, with the columns of the table having the same meaning as in Table 15. Table 18 displays the number "#DT" of the directly-affected trains and "#AT" of trains whose timetable was adjusted. In addition, it contains the specific adjustment applied (number of retimed and rerouted trains), as well as the information on the LR-H algorithm (number of constraints in the constraint pool, number of executed iterations, computing time and percentage optimality gap). In both tables, "Re-R" (Yes or No) represents whether the rerouting strategy is allowed or not in train rescheduling.

Table 17: Results (objective function terms) under different rescheduling strategies (rerouting or not).

Scenario	Re-R	F	f_1	*Per.*	f_2(min)	*Avg.*(min)	#DP
ω9	Yes	223516	699	0.26%	153616	35.32	4349
	No	418520	0	0%	418520	51.28	8162
ω10	Yes	369825	1222	0.45%	247625	27.97	8853
	No	909267	0	0%	909267	77.36	11754
ω11	Yes	510789	1869	0.69%	323889	40.16	8064
	No	1560525	0	0%	1560525	123.03	12684
ω12	Yes	622355	2334	0.86%	388955	40.37	9634
	No	2441820	0	0%	2441820	137.60	17746
ω13	Yes	800913	2539	0.94%	547013	37.06	14762
	No	3584816	0	0%	3584816	157.89	22704

From Table 17 we can see that, when the rerouting strategy is not allowed, although no passengers are abandoned, the total delay increases significantly compared to the case that considers rerouting, and, the longer the duration is, the more it increases. When rerouting is not allowed, the main reason causing large delays is that trains wait for a long time at a station or track before the disrupted area until the disruption is over, thus, not only affecting the passengers at the current station, but also those at all subsequent stations. Therefore, the rerouting strategy is capable of reducing delays, keeping the number of abandoned passengers at a very low percentage.

Table 18: Results (LR-H performance) under different rescheduling strategies (rerouting or not).

Scenario	Re-R	#DT	#AT	#Re-T	#Re-R	Cons.	Iter.	CPU time	GAP
$\omega 9$	Yes	12	19	12	7	39625	37	189s	0.98%
	No	12	25	25	0	37610	14	65s	0.92%
$\omega 10$	Yes	18	36	25	11	47620	50	299s	2.69%
	No	18	26	26	0	47183	48	254s	0.97%
$\omega 11$	Yes	20	35	20	15	56097	50	300s	3.26%
	No	20	32	32	0	49310	50	343s	1.92%
$\omega 12$	Yes	27	46	28	18	60306	50	313s	4.81%
	No	27	45	45	0	54066	50	291s	6.97%
$\omega 13$	Yes	30	55	35	20	67898	50	325s	6.42%
	No	30	45	45	0	60368	50	467s	9.42%

From Table 18, we can see that the problem that allows train rerouting is more complex: indeed, more constraints are added to the constraint pool in the same number of iterations, and more timetables have to be adjusted. We can also observe that, for scenarios $\omega 12$ and $\omega 13$, which have longer disruption durations, the optimality gap is smaller when rerouting can be applied, which also confirms that rerouting is more important when a long disruption occurs.

As an example, we report, in Table 19, the three timetables of train 75 (that was planned to travel along the disrupted area) in scenario $\omega 10$: "Ori." represents the planned timetable, "No-Re." indicates that the rerouting strategy is not allowed, and "Re." that it can be applied.

Table 19: Comparison of timetables of train 75 (original, without rerouting and with rerouting).

Train	Strategy	Timetable
75	Ori.	...20$(448, 450) \rightarrow 18(470, 470) \rightarrow 17(503, 503) \rightarrow$ **3(517,521)** $\rightarrow 16(541, 543) \rightarrow$ $15(560, 562) \rightarrow 14(575, 577) \rightarrow 13(597, 599) \rightarrow 12(611, 611) \rightarrow 11(628, 631) \rightarrow 10(668)$
75	No-Re.	...20$(448, 600) \rightarrow 18(620, 620) \rightarrow 17(652, 652) \rightarrow$ **3(666,668)** $\rightarrow 16(684, 686) \rightarrow$ $15(700, 702) \rightarrow 14(712, 714) \rightarrow 13(730, 732) \rightarrow 12(743, 743) \rightarrow 11(759, 761) \rightarrow 10(794)$
75	Re.	...20$(448, 450) \rightarrow 42(456, 456) \rightarrow 41(466, 466) \rightarrow 40(478, 478) \rightarrow 19(485, 485)$ $\rightarrow 39(498, 498) \rightarrow 38(505, 505) \rightarrow 1(515, 515) \rightarrow 2(519, 519) \rightarrow$ **3(529,531)** $\rightarrow 16(547, 549) \rightarrow$ $15(563, 565) \rightarrow 14(575, 577) \rightarrow 13(597, 599) \rightarrow 12(611, 611) \rightarrow 11(628, 631) \rightarrow 10(668)$

From Table 19, we can see that, when rerouting is not considered, train 75 has to wait at station 20 until the disruption is over (time 600). On the contrary, having the rerouting option, train 75 can travel along a different path. In the table, we highlighted the arrival and departure times at station 3, since this is the first station, after the disrupted area, that is visited in both cases of rerouting or not the train. As can be seen, when the rerouting strategy is not taken into account, the delay at station 3 is very large and propagates to the subsequent stations. On the contrary, when train 75 is rerouted, its delay at station 3 is rather small, and the train can successfully run as scheduled in the original timetable when it arrives at station 14. This example also shows that it is useful to consider the rerouting strategy in order to reduce passenger delays.

6.2.5 Numerical experiments with/without dynamic constraint-generation technique

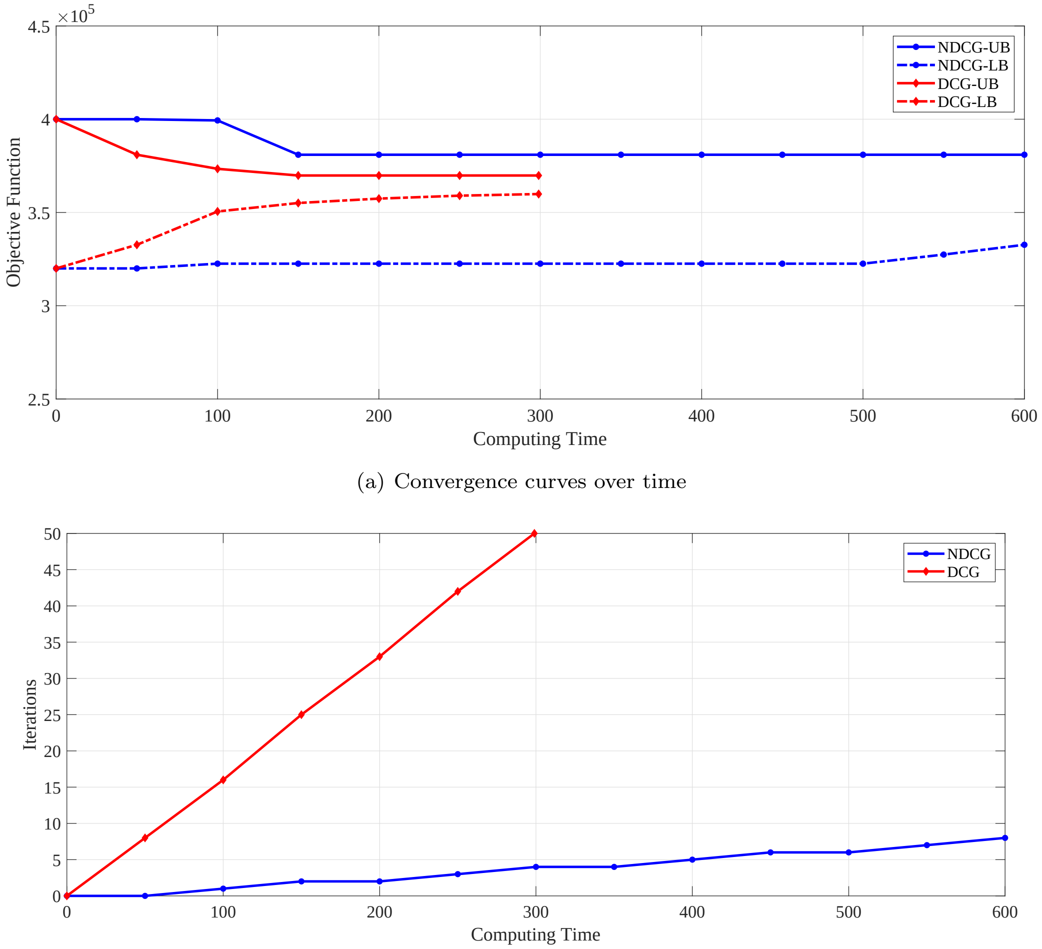
To further highlight the importance of the dynamic constraint-generation technique for accelerating the solution speed, we also execute the LR-H method without applying dynamic constraint-generation. In these experiments, we consider seven scenarios (i.e., $\omega 9$ - $\omega 15$), and compare the results with and without dynamic constraint-generation. Table 20 reports the results, in which "Time(avg.)" represents the average computing time per iteration.

From Table 20 we can see that, without the dynamic constraint-generation technique, the LR-H method requires much longer computing time (10 times longer) to execute one iteration. The reason is that the number of relaxed constraints (6) and (7) is very large, and most of the relaxed constraints are not active during every iteration. When dynamic constraint-generation technique is not applied, many useless calculations are performed, resulting in longer computing time. For clarity, we plot the trend of

Table 20: Comparative results with/without dynamic constraint-generation technique.

Scenario	With dynamic constraint-generation					Without dynamic constraint-generation				
	F	GAP	Iter.	CPU time	Time(avg.)	F	GAP	Iter.	CPU time	Time(avg.)
$\omega 9$	223516	0.98%	37	189s	5.11s	240106	16.25%	8	600s	75.00s
$\omega 10$	369825	2.69%	50	299s	5.98s	380960	12.67%	8	600s	75.00s
$\omega 11$	510789	3.26%	50	300s	6.00s	510789	10.03%	8	600s	75.00s
$\omega 12$	622355	4.81%	50	313s	6.26s	633565	12.39%	8	600s	75.00s
$\omega 13$	800913	6.42%	50	325s	6.50s	835207	15.26%	8	600s	75.00s
$\omega 14$	569457	2.00%	50	178s	3.56s	572644	4.73%	11	600s	54.55s
$\omega 15$	855173	2.62%	50	303s	6.06s	864196	5.82%	11	600s	54.55s
Avg.		3.25%		272s	5.64s		11.02%		600s	69.16s

changes in the best upper bounds, best lower bounds and iterations as the computing time increases, as shown in Figure 16, in which "NDCG" and "DCG" represent not employing and employing the dynamic constraint-generation techniques, respectively.



(a) Convergence curves over time

(b) Curve of the number of iterations over time

Figure 16: Variation of the best upper and lower bounds, and iterations for instance $\omega 10$.

From Figure 16, it can be seen that without the dynamic constraint-generation technique, the LR-H method can only perform less than 10 iterations within the timelimit, i.e., 600s. However, when employing the dynamic constraint-generation technique, the LR-H method can successfully perform 50 iterations within 300 seconds. Therefore, the latter obtains much better lower and upper bounds. This shows that the dynamic constraint-generation technique is very effective in real-time management problems, especially for large-scale problems.

7 Conclusion and Future Work

In this paper, we studied a Train Rescheduling Problem in a large-scale network when a large disruption occurs, including the option of train rerouting. We defined an Integer Linear Programming (ILP) model, based on a space-time network with the goal of minimizing the number of abandoned passengers and the total passenger delay. We developed an efficient heuristic algorithm LR-H that applies Lagrangian relaxation of a subset of constraints in the ILP model to decompose the problem into a series of independent train-based smaller problems. We applied this decomposition and dynamic constraint-generation to solve the problem in short computing time.

The LR-H algorithm was firstly tested on a small-size network in order to compare the obtained results with those found by GUROBI when solving the ILP model: we found that LR-H is able to find near-optimal solutions and very similar adopted train rescheduling for all cases with shorter computing times than GUROBI. Then we applied the LR-H algorithm to a large-scale railway network in China with different disruption scenarios of duration up to 360 minutes, and different computing time limits (from 1 min to 10 min). We found that high quality solutions are obtained within 60 seconds of computing time, and improved optimality gaps (on average 2.27%) are obtained in an average computing time of about 300 seconds. Afterwards, we showed the advantages of employing the rerouting strategy to reduce passenger delays, that are especially evident when disruptions last for a long time. Finally, we validated the effectiveness of applying dynamic constraint-generation to solve large-scale real-time management problems.

The following two aspects can be studied as future research: (1) we considered static passenger demand, i.e., passenger rerouting was not included in the proposed model and in the LR-H, while it could be a useful extension to deal with dynamic passenger rerouting in order to further reduce the number of abandoned passengers; (2) in practice, the duration of a disruption is often uncertain: therefore, it would be interesting to take this uncertainty into account in the solution method.

Acknowledgments

The research was supported by National Natural Science Foundation of China (Nos. 72171023, 71771016, 71825004, 72001019 and 71621001), the program of China Scholarship Council (No. 202007090101) and the Beijing Institute of Technology Research Fund Program for Young Scholars.

References

Altazin, E., Dauzère-Pérès, S., Ramond, F., Tréfond, S., 2020. A multi-objective optimization-simulation approach for real time rescheduling in dense railway systems. European Journal of Operational Research 286, 662–672.

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

Binder, S., Maknoon, Y., Bierlaire, M., 2017. The multi-objective railway timetable rescheduling problem. Transportation Research Part C: Emerging Technologies 78, 78–94.

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

Cadarso, L., Marín, Á., Maróti, G., 2013. Recovery of disruptions in rapid transit networks. Transportation Research Part E: Logistics and Transportation Review 53, 15–33.

Caprara, A., Fischetti, M., Toth, P., 2002. Modeling and solving the train timetabling problem. Operations Research 50, 851–861.

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. Transportation Research Part E: Logistics and Transportation Review 105, 213–239.

Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M., 2010. A tabu search algorithm for rerouting trains during rail operations. Transportation Research Part B: Methodological 44, 175–192.

D'Ariano, A., Corman, F., Pacciarelli, D., Pranzo, M., 2008. Reordering and local rerouting strategies to manage train traffic in real time. Transportation Science 42, 405–419.

D'Ariano, A., Meng, L., Centulio, G., Corman, F., 2019. Integrated stochastic optimization approaches for tactical scheduling of trains and railway infrastructure maintenance. Computers & Industrial Engineering 127, 1315–1335.

D'Ariano, A., Pacciarelli, D., Pranzo, M., 2007a. A branch and bound algorithm for scheduling trains in a railway network. European Journal of Operational Research 183, 643–657.

D'Ariano, A., Pranzo, M., Hansen, I.A., 2007b. Conflict resolution and train speed coordination for solving real-time timetable perturbations. IEEE Transactions on Intelligent Transportation Systems 8, 208–222.

Dollevoet, T., Huisman, D., Kroon, L., Schmidt, M., Schöbel, A., 2015. Delay management including capacities of stations. Transportation Science 49, 185–203.

Dollevoet, T., Huisman, D., Kroon, L.G., Veelenturf, L.P., Wagenaar, J.C., 2016. Application of an iterative framework for real-time railway rescheduling. Computers & Operations Research 78, 203–217.

Dollevoet, T., Huisman, D., Schmidt, M., Schöbel, A., 2012. Delay management with rerouting of passengers. Transportation Science 46, 74–89.

Fang, W., Yang, S., Yao, X., 2015. A survey on problem models and solution approaches to rescheduling in railway networks. IEEE Transactions on Intelligent Transportation Systems 16, 2997–3016.

Fisher, M.L., 1981. The lagrangian relaxation method for solving integer programming problems. Management Science 27, 1–18.

Gao, Y., Yang, L., Gao, Z., 2017. Real-time automatic rescheduling strategy for an urban rail line by integrating the information of fault handling. Transportation Research Part C: Emerging Technologies 81, 246–267.

Ghaemi, N., Cats, O., Goverde, R., 2017. A microscopic model for optimal train short-turnings during complete blockages. Transportation Research Part B: Methodological 105, 423–437.

Ghaemi, N., Cats, O., Goverde, R., 2018a. Macroscopic multiple-station short-turning model in case of complete railway blockages. Transportation Research Part C: Emerging Technologies 89, 113–132.

Ghaemi, N., Zilko, A.A., Yan, F., Cats, O., Kurowicka, D., Goverde, R., 2018b. Impact of railway disruption predictions and rescheduling on passenger delays. Journal of Rail Transport Planning & Management 8, 103–122.

Hong, X., Meng, L., D'Ariano, A., Veelenturf, L.P., Long, S., Corman, F., 2021. Integrated optimization of capacitated train rescheduling and passenger reassignment under disruptions. Transportation Research Part C: Emerging Technologies 125, 103025.

Jespersen-Groth, J., Potthoff, D., Clausen, J., Huisman, D., Kroon, L.G., Maróti, G., Nielsen, M.N., 2009. Disruption management in passenger railway transportation. In: Ahuja, R.K., Möhring, R.H., Zaroliagis, C.D. (Eds.), Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems. Springer Berlin Heidelberg , pp. 399–421.

Lamorgese, L., Mannino, C., 2015. An exact decomposition approach for the real-time train dispatching problem. Operations Research 63, 48–64.

Lamorgese, L., Mannino, C., Pacciarelli, D., Krasemann, J.T., 2018. Train dispatching. In: Handbook of Optimization in the Railway Industry, International Series in Operations Research & Management Science 268, 265–283.

Lamorgese, L., Mannino, C., Piacentini, M., 2016. Optimal train dispatching by benders'-like reformulation. Transportation Science 50, 910–925.

Louwerse, I., Huisman, D., 2014. Adjusting a railway timetable in case of partial or complete blockades. European Journal of Operational Research 235, 583–593.

Mascis, A., Pacciarelli, D., 2002. Job-shop scheduling with blocking and no-wait constraints. European Journal of Operational Research 143, 498–517.

Meng, L., Zhou, X., 2011. Robust single-track train dispatching model under a dynamic and stochastic environment: A scenario-based rolling horizon solution approach. Transportation Research Part B: Methodological 45, 1080–1102.

Meng, L., Zhou, X., 2014. Simultaneous train rerouting and rescheduling on an N-track network: A model reformulation with network-based cumulative flow variables. Transportation Research Part B: Methodological 67, 208–234. doi:https://doi.org/10.1016/j.trb.2014.05.005.

Narayanaswami, S., Rangaraj, N., 2013. Modelling disruptions and resolving conflicts optimally in a railway schedule. Computers & Industrial Engineering 64, 469–481.

Pellegrini, P., Marlière, G., Rodriguez, J., 2014. Optimal train routing and scheduling for managing traffic perturbations in complex junctions. Transportation Research Part B: Methodological 59, 58–80.

Pellegrini, P., Pesenti, R., Rodriguez, J., 2019. Efficient train re-routing and rescheduling: Valid inequalities and reformulation of recife-milp. Transportation Research Part B: Methodological 120, 33–48.

Schachtebeck, M., Schöbel, A., 2010. To wait or not to wait-and who goes first? Delay management with priority decisions. Transportation Science 44, 307–321.

Schöbel, A., 2007. Integer programming approaches for solving the delay management problem. In: Algorithmic methods for railway optimization. Springer , pp. 145–170.

Schöbel, A., 2009. Capacity constraints in delay management. Public Transport 1, 135–154.

Veelenturf, L.P., Kidd, M.P., Cacchiani, V., Kroon, L.G., Toth, P., 2016. A railway timetable rescheduling approach for handling large-scale disruptions. Transportation Science 50, 841–862.

Veelenturf, L.P., Kroon, L.G., Maroti, G., 2017. Passenger oriented railway disruption management by adapting timetables and rolling stock schedules. Transportation Research Part C: Emerging Technologies 80, 133–147.

Wang, Y., Zhao, K., D'Ariano, A., Niu, R., Li, S., Luan, X., 2021. Real-time integrated train rescheduling and rolling stock circulation planning for a metro line under disruptions. Transportation Research Part B: Methodological 152, 87–117.

Xu, P., Corman, F., Peng, Q., 2016. Analyzing railway disruptions and their impact on delayed traffic in Chinese high-speed railway. IFAC-PapersOnLine 49, 84–89.

Zhan, S., Kroon, L., Veelenturf, L., Wagenaar, J., 2015. Real-time high-speed train rescheduling in case of a complete blockage. Transportation Research Part B: Methodological 78, 182–201.

Zhan, S., Kroon, L., Zhao, J., Peng, Q., 2016. A rolling horizon approach to the high speed train rescheduling problem in case of a partial segment blockage. Transportation Research Part E: Logistics and Transportation Review 95, 32–61.

Zhan, S., Wong, S., Shang, P., Peng, Q., Xie, J., Lo, S., 2021. Integrated railway timetable rescheduling and dynamic passenger routing during a complete blockage. Transportation Research Part B: Methodological 143, 86–123.

Zhang, C., Gao, Y., Yang, L., Gao, Z., Qi, J., 2020. Joint optimization of train scheduling and maintenance planning in a railway network: A heuristic algorithm using lagrangian relaxation. Transportation Research Part B: Methodological 134, 64–92.

Zhang, H., Li, S., Wang, Y., Wang, Y., Yang, L., 2021. Real-time optimization strategy for single-track high-speed train rescheduling with disturbance uncertainties: A scenario-based chance-constrained model predictive control approach. Computers & Operations Research 127, 105135.

Zhang, Q., Lusby, R.M., Shang, P., Zhu, X., 2022. A heuristic approach to integrate train timetabling, platforming, and railway network maintenance scheduling decisions. Transportation Research Part B: Methodological 158, 210–238.

Zhang, Y., Peng, Q., Yao, Y., Zhang, X., Zhou, X., 2019. Solving cyclic train timetabling problem through model reformulation: Extended time-space network construct and alternating direction method of multipliers methods. Transportation Research Part B: Methodological 128, 344–379.

Zhu, Y., Goverde, R., 2017. System-based vulnerability measures for railway systems. The 7th International Conference on Railway Operations Modelling and Analysis (RailLille2017) .

Zhu, Y., Goverde, R., 2019. Railway timetable rescheduling with flexible stopping and flexible short-turning during disruptions. Transportation Research Part B: Methodological 123, 149–181.

Zhu, Y., Goverde, R., 2020a. Dynamic and robust timetable rescheduling for uncertain railway disruptions. Journal of Rail Transport Planning & Management 15, 100196.

Zhu, Y., Goverde, R., 2020b. Integrated timetable rescheduling and passenger reassignment during railway disruptions. Transportation Research Part B: Methodological 140, 282–314.

Zhu, Y., Goverde, R., 2021. Dynamic railway timetable rescheduling for multiple connected disruptions. Transportation Research Part C: Emerging Technologies 125, 103080.