Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Emergency Train Scheduling on Chinese High-Speed Railways

Yu-Jun Zheng

To cite this article:
Yu-Jun Zheng (2018) Emergency Train Scheduling on Chinese High-Speed Railways. Transportation Science 52(5):1077-1091. https://doi.org/10.1287/trsc.2017.0794

informs
http://pubsonline.informs.org/journal/trsc/
TRANSPORTATION SCIENCE
Vol. 52, No. 5, September–October 2018, pp. 1077–1091
ISSN 0041-1655 (print), ISSN 1526-5447 (online)

Emergency Train Scheduling on Chinese High-Speed Railways

Yu-Jun Zheng[a]

[a] College of Computer Science and Technology, Zhejiang University of Technology, 310023 Hangzhou, China
Contact: yujun.zheng@computer.org, http://orcid.org/0000-0002-6095-6325 (Y-JZ)

Received: April 8, 2016
Revised: November 16, 2016; April 3, 2017; June 21, 2017
Accepted: July 2, 2017
Published Online in Articles in Advance: November 1, 2017

https://doi.org/10.1287/trsc.2017.0794

Abstract. China has recently developed a large-scale high-speed railway (HSR) network that not only revolutionizes the perception of long-haul transport but also fundamentally improves the emergency response capability of the country. This gives rise to a problem of scheduling a set of emergency trains within the predefined HSR timetable. According to different degrees of disturbance of the timetable, this paper identifies three kinds of solutions, named, ideal, quasi-ideal, and feasible solutions, to the problem. We define a "headroom" function for computing the maximum number of emergency trains that can be inserted between two adjacent regular trains without any disturbance, from which a linear algorithm for finding ideal solutions is derived. By identifying the scope of changes to each regular train, quasi-ideal and feasible solutions can be found based on the ideal solutions on a set of alternative timetables. Numerical experiments demonstrate that our method is very efficient for supporting near-immediate emergency responses. We have used the solution method to evaluate the emergency capacity of most HSR lines in China. In particular, we successfully applied the method to schedule emergency trains in the rescue operation of the 2015 Tianjin chemical explosion.

Funding: This work was supported by the National Natural Science Foundation of China under [Grant 61473263].
Supplemental Material: The online appendix is available at https://doi.org/10.1287/trsc.2017.0794.

Keywords: high-speed railway (HSR) • railway transportation • emergency transportation • train timetabling

1. Introduction

The average speed of conventional trains in China was about 60 km/hr in the 1990s and did not exceed 150 km/hr in the early 2000s, which had a significant negative impact on the economic exchange and development of the country for a long time. The government started its ambitious high-speed railway (HSR) plan in 2006. The first set of bullet trains with a maximum speed of 250 km/hr operated on some existing or updated lines in 2007, and the first HSR line with a maximum speed of 350 km/hr opened between Beijing and Tianjin in 2008 to coincide with the 2008 Beijing Olympic Games. Now the HSR network in China is over 22,000 km long and carries more than four million passengers a day. It is observed that HSR has greatly shortened the time-space distances between megacities and stimulated the development of second- and third-tier cities (Zheng and Kahn 2013), and is expected to have some significant macro effects on the national pattern of activity (Cao et al. 2013).

In addition, one reason the government built the HSR network is to fundamentally improve its emergency response capability. In fact, China's railway transportation has played a dominant role in disaster relief, military, and other emergency supply chains in China for decades (Zheng and Ling 2013; Zheng et al. 2014b, 2015b). In comparison with conventional trains, bullet trains can deliver personnel, equipment, and supplies in a much more efficient way to the affected areas that are too far to reach by car and cannot meet the requirements for aircraft operations.

The research on the models and methods for scheduling and rescheduling of trains is extensive (Cordeau, Toth, and Vigo 1998; Caprara, Fischetti, and Toth 2002; Lusby et al. 2011; Wagenaar, Kroon, and Schmidt 2017), but few have been conducted on HSR train scheduling for temporary emergency delivery. Emergency HSR train scheduling can be a challenging task because of the following reasons:

- The high train speed and high train frequency of HSR.
- The relatively large number of emergency trains to be accommodated, especially for large-scale relief or military operations.
- The high sensitivity of HSR timetables, i.e., the disturbance to HSR timetables, including direct and knock-on delays to regular trains (Higgins and Kozan 1998;

Khadilkar 2017), can cause significant direct and indirect economic losses to the HSR holder.

In this paper, we consider the problem of rescheduling a predefined HSR timetable to accommodate a set of required emergency trains, the aim of which is to make the emergency trains reach the destination as quickly as possible, while not disturbing the regular train schedules too much. Specifically, given a predefined HSR timetable and a set of emergency trains, we are interested in the following three questions (Q1–Q3):

Q1. Can we (and if so, how can we) insert the emergency trains into the timetable without disturbing the schedule of any regular train?

Q2. Can we (and if so, how can we) insert the emergency trains into the timetable without delaying any regular train?

Q3. Can we (and if so, how can we) insert the emergency trains into the timetable without violating the specified delay constraint of any regular train?

Remark 1. We use the term "disturbing" to indicate any change to a predefined schedule and the term "delaying" to indicate any later arrival at planned stops—the former is from the point of view of train operators, while the latter is from that of passengers. For example, a solution to Q2 allows that a train to arrive later at some nonstop stations as long as it can arrive at each planned stop station on time, but a solution to Q1 does not allow this. The motivation lies in the fact that in Chinese HSR, most regular trains run at a lower speed (250~300 km/hr), and in emergency situations they can give in to emergency trains and then run at a much higher speed (e.g., 350 km/hr) to recover the delay.

To answer the above questions, we develop a solution method based on time-space networks, where each train schedule is represented by a flow line. The key to our method is the definition of a "headroom" function for computing the maximum number of emergency trains that can be inserted between two adjacent flow lines without any disturbance, which directly supports solutions to Q1. For Q2 and Q3 that allow changing regular flow lines, we identify all allowable alternative timetables, and then obtain the optimal solution to Q2 or Q3 based on the solutions to Q1 on those alternative timetables. Hence the main outcome of this paper includes a linear algorithm for solving Q1 and two efficient branch-and-bound algorithms for Q2 and Q3, respectively. The contribution of our method can be summarized as follows:

- It can produce solutions to the above questions directly and efficiently (within 20 seconds for almost all of the HSR lines in China, as shown in the numerical experiments in Section 5) and thus support near-immediate emergency responses.
- It provides a convenient way to evaluate the capacity of an HSR line for emergency trains, i.e., the maximum number of emergency trains that can be inserted into the regular timetable, which is of critical importance for emergency operation planning.
- Retroactively, it offers a way to reserve adequate space in a regular HSR timetable for accommodating emergency trains, which is also important in train timetabling.
- It has the potential to be generalized or extended for a wide range of seemingly different problems considering the accommodation of a set of temporary services in a predefined timetable, where services and resources can be analogized as trains and sections.

In practice, we used the method to evaluate the "emergency capacity" of almost all HSR lines in China, and applied it to the rescue operation in the 2015 Tianjin chemical explosion.

In the rest of the paper, we review the literature in Section 2, present the problem model in Section 3, and propose our solution method in Section 4. Section 5 presents the numerical experiments, Section 6 describes the real-world application, and Section 7 concludes.

2. Literature Review

Train timetabling (scheduling) is a difficult task of vital importance. Train timetables were mainly constructed manually before the 1990s, and the planner was often satisfied with a feasible timetable rather than an optimized one (Brannlund et al. 1998). In the past two decades, computerized scheduling systems have played a dominant role in complex timetabling. As a subclass of scheduling problems, timetabling problems are often formulated based on periodic event scheduling (Serafini and Ukovich 1989; Kroon and Peeters 2003), job shop scheduling (Odijk 1996; Liu and Kozan 2011), and resource-constrained project scheduling (Zhou and Zhong 2005) models, almost all of which are known to be NP-hard (Cordeau, Toth, and Vigo 1998). Classical enumeration-based methods with guaranteed optimality, such as branch-and-bound (Jovanovic and Harker 1991; Higgins, Kozan, and Ferreira 1996; D'Ariano, Pacciarelli, and Pranzo 2007; Zhou and Zhong 2007), backtracking (Voorhoeve 1993), and cut generation (Odijk 1996), only perform well on instances of small or moderate size. As the size and complexity of timetables increase, many researchers have been working on heuristic/metaheuristic methods, including local search (Higgins, Kozan, and Ferreira 1997; Burdett and Kozan 2010), Lagrangian heuristics (Brannlund et al. 1998; Caprara, Fischetti, and Toth 2002; Caprara et al. 2006), greedy heuristics (Cai, Goh, and Mees 1998; Dorfman and Medanic 2004), and evolutionary algorithms (Kwan and Mistry 2003; Semet and Schoenauer 2005; Xu et al. 2014; Zheng, Ling, and Xue 2014a; Zheng, Zhang, and Zhang 2016), which can produce near-optimal solutions within a reasonable amount of time.

The aforementioned models and solution methods mainly focus on the development of new timetables that are typically (but not necessarily) to be applied periodically (Burdett and Kozan 2010). There is another variant that concerns the redevelopment of existing timetables. Sahin (1999) studies a problem of rescheduling trains to solve intertrain conflicts caused by disruptions in a single-track railway, and develops a heuristic algorithm that detects the trains involved in the immediate conflict and stops each train sequentially for the resolution, and then uses a look-ahead method to search for the consequences by considering potential conflicts and associated delays. Adenso-Diaz, Gonzalez, and Gonzalez-Torre (1999) consider a similar rescheduling problem for solving the distortion of established timetables and develop a backtracking-based heuristic where each solution is evaluated based on the service priority, the passengers transported, and the delays. Burdett and Kozan (2009) characterize a problem of inserting additional trains as a hybrid job shop scheduling problem, which is solved using a constructive algorithm and metaheuristics. Cacchiani, Caprara, and Toth (2010) solve a similar problem of scheduling extra freight trains by a Lagrangian heuristic. The rescheduling algorithm by Krasemann (2012) performs a depth-first search to produce a first feasible solution and then branches to find better solutions, but the second phase can be inefficient. Albrecht, Panton, and Lee (2013) present a model for simultaneously minimizing the delays of train arrival and track maintenance, which is solved by first constructing schedules for trains not yet at the destination and then selecting the next train movement based on the earliest possible start time. Dündar and Şahin (2013) use a genetic algorithm (GA) for train rescheduling based on a conflict-free genetic representation where each bit contains information on train pairs' meeting and passing order over the line. Isaai and Cassaigne (2001) study both predictive and reactive train scheduling to solve disruptions to timetables with imprecise plans, and they combine the use of data and knowledge and the differences in uncertainty levels to compare the solutions. Yang, Zhou, and Gao (2014) use another credibility-based way to capture the uncertainty of incident duration based on empirical judgments and then to seek a reliable rescheduling plan. Considering a complete blockage of a railway segment for a certain period of time, Zhan et al. (2016) present a real-time timetable rescheduling problem, which is solved by a mixed-integer programming method to minimize the total weighted train delay and the number of canceled trains, and test their approach on instances of the Beijing–Shanghai HSR line. The rescheduling problem is further extended by Veelenturf et al. (2016) to handle large-scale disruptions and is tested on the Dutch railway network.

The work on temporary scheduling of emergency trains within a predefined timetable, though of practical importance, is scarce in the literature. Such a scheduling problem can be hard in an HSR line because of the high train speed and frequency, and can be harder in a line carrying both high-speed and traditional trains. The aim of this paper is to develop an efficient solution method to the problem.

3. The Problem Statement

3.1. Model Assumptions

In our model, we view an HSR line as a sequence of block sections (or segments) divided by signals. At any time, there cannot be more than one train traveling in the same direction in a section. For simplicity, we view each station as a junction of sections, i.e., the length of the station is regarded as zero (and the actual length can be added to the associated sections), as shown in Figure 1.

The formulation of the HSR train scheduling problem is based on the following assumptions A1–A3:

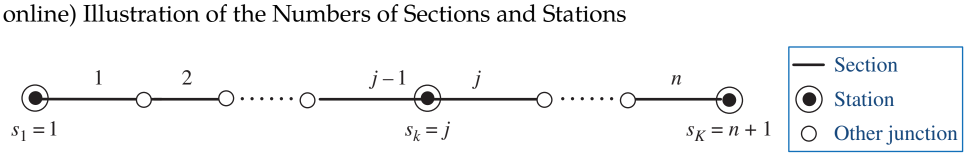
A1. The regular trains can have the same or different maximum velocities, but all of the emergency trains have the same maximum velocity, which is not less than any regular train.

A2. There is no overtaking among the emergency trains (in the rest of the paper, the term "overtaking" refers to an emergency train overtaking a regular train unless otherwise specified).

A3. All of the sections are double tracked: an up train cannot enter into the down track, and vice versa. However, the number of trains that can reside in a station section is unlimited.

Remark 2. In practice, the track number of a station section is certainly limited, but we do not consider this limitation as an input to the problem. Instead we regard it as a postcondition, i.e., for each solution we check whether the number of trains residing in a station at a time will exceed the number of tracks. As far as we can see from our extensive experiments, this never happens.

Figure 1. (Color online) Illustration of the Numbers of Sections and Stations



For ease of discussion, the following assumptions are temporarily made for the problem (in Section 4.4 we will show how to handle the cases where these assumptions do not hold):

Å1. There is no overtaking among the regular trains in the original timetable.

Å2. All regular trains travel from the first (original) station to the last (terminal) station.

3.2. Input, Decision, and Intermediate Variables

Formally, the input of the problem can be divided into two parts. The first part is a predefined timetable, denoted by X, which consists of the following components:

- I The set of m regular trains.
- J The sequence of n sections.
- S The sequence of K stations.
- d_j The length of section j ($j \in J$).
- $v^{\mathrm{U}}_{i,j}$ The maximum allowable velocity of train i on section j ($i \in I; j \in J$).
- $t_{i,j}$ The predefined time of train i entering section j ($i \in I; j \in J$).
- $\hat{t}_{i,s}$ The predefined time of train i arriving at station s ($i \in I; s \in S$).
- $\dot{t}_j$ The latest time of section j for passing trains ($j \in J$).
- $\tilde{t}$ The (minimum) buffer time between a train leaving a section and the next train entering the section.

Excluding the original station and the terminal station, each intermediate station is a junction of two adjacent sections (but the opposite is not necessarily true). For notational simplicity, each nonterminal station is denoted as the number of the section after the station, and the terminal station is denoted as $n+1$, as illustrated by Figure 1.

In the present problem, each train i has an elastic time interval constrained by d_j and $v^{\mathrm{U}}_{i,j}$ for passing through each section j, and $t_{i,j+1}$ is also the predefined time for train i leaving section j. For each $s \in S$ we have $t_{i,s} \geq \hat{t}_{i,s}$, and the equality holds when train i does not stop at station s.

The second part of the problem input is a set E of p temporary emergency trains to be inserted into the predefined timetable X (here the term "insertion" indicates that we only consider the emergency trains that depart later than the first regular train; the number of emergency trains that can be scheduled before the first regular train is easy to determine and thus is excluded from the considered problem). At each section j, there is also an upper velocity limit, denoted by v^{U}_j, which is the same for all emergency trains. For Q3 there is also a latest time $\ddot{t}_i$ of each regular train i arriving at the terminal station.

A solution to the problem is a new timetable (denoted by X') of the train set $I' = I \cup E$, which includes the following decision variables for all $i \in I'$, $j \in J$, and $s \in S$:

- $t^{\mathrm{R}}_{i,j}$ The real entry time of train i into section j in X'.
- $\hat{t}^{\mathrm{R}}_{i,s}$ The real arrival time of train i at station s in X'.

To evaluate the disturbance of the predefined timetable, we calculate the following intermediate variables from the input and/or decision variables:

$\Delta t_{i,s} = t_{i,s} - \hat{t}_{i,s}$:
The predefined dwell time of train i at station s in X.

$S_i = \{s \in S \mid \Delta t_{i,s} > 0\}$:
The set of stations that train i plans to stop at in X.

$\Delta t^{\mathrm{R}}_{i,s} = t^{\mathrm{R}}_{i,s} - \hat{t}^{\mathrm{R}}_{i,s}$:
The real dwell time of train i at station s in X'.

$\nabla \hat{t}_{i,s} = \max(\hat{t}^{\mathrm{R}}_{i,s} - \hat{t}_{i,s}, 0)$:
The delay of the arrival time of train i at station s.

For notational simplicity we will use $t_{e,j}$ to denote $t^{\mathrm{R}}_{e,j}$ if e definitely denotes an emergency train. We also use $\tau_{i,j}$ and $\tau^{\mathrm{R}}_{i,j}$ to denote the predefined released time and real released time of section j by train i

$$\tau_{i,j} = \begin{cases} \hat{t}_{i,j+1}, & \text{if } (j+1) \in S \\ t_{i,j+1}, & \text{else}, \end{cases}$$

$$\tau^{\mathrm{R}}_{i,j} = \begin{cases} \hat{t}^{\mathrm{R}}_{i,j+1}, & \text{if } (j+1) \in S \wedge \Delta t^{\mathrm{R}}_{i,j+1} > 0 \\ t^{\mathrm{R}}_{i,j+1}, & \text{else}. \end{cases}$$

3.3. Constraints and Objectives

To ensure traffic safety, a train cannot enter a section unless the section has been released for at least the buffer time. The predefined timetable already satisfies this requirement, i.e., $|t_{i,j} - \tau_{i',j}| \geq \tilde{t}$ holds for all $i, i' \in I$ and $j \in J$. After insertion, the new timetable must also satisfy the hard constraint

$$|t^{\mathrm{R}}_{i,j} - \tau^{\mathrm{R}}_{i',j}| \geq \tilde{t}, \quad \forall\, i, i' \in I'; \forall\, j \in J. \tag{1}$$

The following three constraints should also be satisfied.

- The travel speed of each train cannot exceed its upper limit

$$t^{\mathrm{R}}_{i,j+1} \geq \begin{cases} t^{\mathrm{R}}_{i,j} + d_j / v^{\mathrm{U}}_{i,j} & i \in I;\, j \in J \backslash \{n\} \\ t^{\mathrm{R}}_{i,j} + d_j / v^{\mathrm{U}}_{j} & i \in E;\, j \in J \backslash \{n\}. \end{cases} \tag{2}$$

- The predefined dwell time of a regular train at a planned stop should not be decreased

$$\Delta t^{\mathrm{R}}_{i,j} \geq \Delta t_{i,j}, \quad \forall\, i \in I; \forall\, j \in S_i. \tag{3}$$

- A regular train cannot leave a planned stop before the predefined time

$$t^{\mathrm{R}}_{i,j} \geq t_{i,j}, \quad \forall\, i \in I; \forall\, j \in S_i. \tag{4}$$

Considering Q1–Q3, we, respectively, identify the following three types of solutions.

Definition 1. An *ideal solution* (for Q1) is a timetable X' where constraints (1) and (2) are satisfied while all regular trains are scheduled as in the predefined timetable (and thus constraints (3) and (4) will be automatically satisfied)

$$t^{R}_{i,j} = t_{i,j}, \quad \forall\, i \in I; \forall\, j \in J, \tag{5}$$
$$\hat{t}^{R}_{i,j} = \hat{t}_{i,j}, \quad \forall\, i \in I; \forall\, j \in S_i. \tag{6}$$

Definition 2. A *quasi-ideal solution* (for Q2) is a timetable X' where constraints (1)–(4) are satisfied while no regular train will be delayed on any planned stop

$$\hat{t}^{R}_{i,j} \le \hat{t}_{i,j}, \quad \forall\, i \in I; \forall\, j \in S_i. \tag{7}$$

Definition 3. A *feasible solution* (for Q3) is a timetable X' where constraints (1)–(4) are satisfied while the real arrival time of each regular train at the terminal station should not be later than its deadline

$$\hat{t}^{R}_{i,n+1} \le \ddot{t}_i, \quad \forall\, i \in I. \tag{8}$$

To answer Q1, Q2, or Q3, we are to determine whether there is an ideal, quasi-ideal, or feasible solution for inserting E into X and, if so, produce such a solution that makes each emergency train in E reach the destination as early as possible

$$\min \sum_{i\in E} \hat{t}^{R}_{i,n+1}. \tag{9}$$

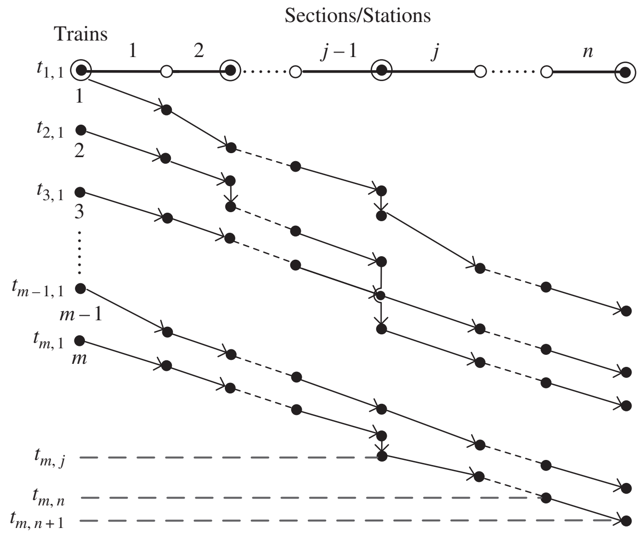
For simplicity, we present the complete mathematical programming formulation in Online Appendix A. The answer to Q1–Q3 is yes if we can find a solution optimizing the objective function while satisfying all of the constraints and is no otherwise. If a problem has a solution when $|E| = p$ but has no solution when $|E| = p + 1$ (while all other inputs remain the same), then p is the maximum number of emergency trains that can be inserted into the predefined timetable.

3.4. A Time-Space Network Representation

For the considered problem, a time-space network (Chardaire et al. 2005) can be used to represent the movement of trains over the time horizon. As illustrated in Figure 2, the vertical axis (time dimension) represents the time duration, the horizontal axis (space dimension) represents the sections, each flow node denotes the time point that a train enters a section or arrives at a station, and the route of each train is a flow line that connects the nodes of the train in time sequence.

Remark 3. We are only concerned with the time points on the junctions of the sections, thus, we always draw a straight line to connect two nodes no matter whether the train passes the section with a *constant velocity*—the smaller the slope of the line, the higher the *average velocity* of the train in the section. According to our problem model, each train has a maximum allowable velocity in each section, hence there is a lower limit on the slope of each line.

Figure 2. Illustration of a Railway Time-Space Network



A vertical line denotes that the train stops at the corresponding station, and the length of the line is proportional to the dwell time. In the railway time-space network, the "blocking" constraint can be expressed as we cannot draw a horizontal line crossing two or more flow lines within (the open interval of) any section. Thereby, the problem of inserting a set of emergency trains into a predefined timetable is equivalent to inserting a set of flow lines into the time-space network of the timetable while satisfying all specified constraints.

4. Solution Method

In this section we develop algorithms for finding solutions for Q1, Q2, and Q3 (according to Definitions 1–3), respectively.

4.1. Ideal Solutions

An ideal solution to the problem inserts a set of flow lines of the emergency trains into the time-space network without modifying any existing flow lines. Depending on whether an emergency train can overtake a regular train, the solutions can be divided into two cases.

4.1.1. Ideal Solutions Without Overtaking. Suppose an emergency train e is to be inserted between two regular trains i and $i+1$ without overtaking i. Based on constraint (1) we have

$$t_{e,j} \ge \tau^{R}_{i,j} + \tilde{t}, \quad j \in J,$$
$$\begin{cases} t^{R}_{i+1,j} \ge t_{e,j+1} + \tilde{t} & (j+1) \notin S_i \\ t^{R}_{i+1,j} \ge \hat{t}_{e,j+1} + \tilde{t} & (j+1) \in S_i. \end{cases}$$

Thus we can derive

$$t^{\mathrm{R}}_{i+1,j} \geq \tau^{\mathrm{R}}_{i,j+1} + 2\tilde{t}, \quad j \in J \wedge j < n, \tag{10}$$

$$\begin{cases} t^{\mathrm{R}}_{i+1,j} - \tau^{\mathrm{R}}_{i,j} \geq t_{e,j+1} - t_{e,j} + 2\tilde{t}, & (j+1) \notin S_i \\ t^{\mathrm{R}}_{i+1,j} - \tau^{\mathrm{R}}_{i,j} \geq \hat{t}_{e,j+1} - t_{e,j} + 2\tilde{t} & (j+1) \in S_i. \end{cases} \tag{11}$$

The velocity limits indicate $\tau_{e,j} - t_{e,j} \geq d_j/v^{\mathrm{U}}_j$, and thus from Equation (11) we can derive

$$t^{\mathrm{R}}_{i+1,j} - \tau^{\mathrm{R}}_{i,j} - \tilde{t} \geq d_j/v^{\mathrm{U}}_j + \tilde{t}, \quad j \in J. \tag{12}$$

Equations (12) and (10) must hold to allow the insertion of an emergency train between trains i and $i+1$. By extension, to enable the insertion of θ emergency trains between i and $i+1$, the following conditions must hold:

$$t^{\mathrm{R}}_{i+1,j} - \tau^{\mathrm{R}}_{i,j} - \tilde{t} \geq \theta(d_j/v^{\mathrm{U}}_j + \tilde{t}), \quad j \in J, \tag{13}$$

$$t^{\mathrm{R}}_{i+1,j} \geq \tau^{\mathrm{R}}_{i,j+\theta} + (\theta+1)\tilde{t}, \quad j \in J \wedge j \leq n-\theta, \tag{14}$$

where Equation (13) indicates that the idle time interval between trains i and $i+1$ on each section is sufficient for θ emergency trains to pass through the section, and Equation (14) indicates that train i should release section $j+\theta$ before train $i+1$ enters section j such that there are θ idle block sections for accommodating θ emergency trains.

Because in an ideal solution $\tau^{\mathrm{R}}_{i,j} = \tau_{i,j}$ and $t^{\mathrm{R}}_{i,j} = t_{i,j}$ always hold, we define the following two threshold functions for all $s, s' \in S$ and $1 \leq i \leq m$:

$$\Upsilon(i,s,s') = \min_{s \leq j < s'} \{(t_{i+1,j} - \tau_{i,j} - \tilde{t})/(d_j/v^{\mathrm{U}}_j + \tilde{t})\}, \tag{15}$$

$$\Theta(i,s,s') = \operatorname*{arg\,max}_{1 \leq \theta \leq \Upsilon(i,s,s')} \{\forall\, j\colon s \leq j < s' - \theta\colon \; t_{i+1,j} \geq \tau_{i,j+\theta} + (\theta+1)\tilde{t}\}. \tag{16}$$

Theorem 1. *The maximum number of nonovertaking emergency trains that can be inserted between regular trains i and $i+1$ from station s to station s' without disturbing the regular trains is $\Theta(i,s,s')$.*

The proof of Theorem 1 is given in Online Appendix B.

We call $\Theta(i,s,s')$ the *headroom* between i and $i+1$ from s to s'. Therefore, if the headroom $\theta_i = \Theta(i,1,n+1)$ is larger than zero, we can iteratively insert θ_i emergency trains between i and $i+1$ without disturbing the other trains. Each emergency flow line e is determined by first initializing

$$t_{e,1} = \tau^{\mathrm{R}}_{i,1} + \tilde{t}, \tag{17}$$

and then iteratively applying

$$t_{e,j+1} = \max(\tau^{\mathrm{R}}_{i,j+1} + \tilde{t}, t_{e,j} + d_j/v^{\mathrm{U}}_j), \quad j \in J \wedge j < n, \tag{18}$$

$$\hat{t}_{e,j+1} = t_{e,j} + d_j/v^{\mathrm{U}}_j, \quad (j+1) \in S, \tag{19}$$

and finally reaching $\hat{t}_{e,n+1} = t_{e,n} + d_n/v^{\mathrm{U}}_n$. Note that i in Equations (17) and (18) will also increase after each insertion, i.e., i always denotes the index of the train after which e will be inserted.

Remark 4. Equations (18) and (19) make every emergency train run at the maximum velocity and thus leaves a maximum headroom after the train. Note that if an emergency train needs to wait at an intermediate station and the time interval between the emergency train and its subsequent train is more than the buffer time, we can derive other ideal solutions by making the emergency train run at a lower velocity in the section before the intermediate station, and in practice such "moderate" solutions might be more preferable. The derivation of alternative solutions from our ideal solutions is simple and is omitted.

Theorem 1 and Equations (17)–(19) lead to Algorithm 1 for finding an ideal solution to insert a set of p emergency trains into a predefined timetable X. Calculation of the headroom for each i requires $O(n)$ time, and thus Algorithm 1 has a worst case time complexity of $O(n(m+p))$. However, since there are $T/\tilde{t}$ choices for determining the entry time of each emergency train at each section (where T denotes the total time horizon), the complexity of a standard integer programming method for solving the problem is $O((T/\tilde{t})^{pn})$, which is considerably higher than our method. The algorithms that follow for finding quasi-ideal and feasible solutions is based on this linear algorithm, and thus are typically much faster than the corresponding integer programming methods.

Algorithm 1 (Searching for an ideal solution without overtaking)

Input A predefined timetable $X = (I, J, S, d[n], v^{\mathrm{U}}[m,n], t[m+1,n], \hat{t}[m+1,K], \tilde{t})$, the number p of the emergency trains, and the maximum velocity vector $v^{\mathrm{U}}[n]$

Output An ideal timetable X'

```
let X' = X;
for i = 1 to m do
  γ_i ← min_{1≤j≤n} {(t_{i+1,j} − τ_{i,j} − t̃)/(d_j/v^U_j + t̃)};
  θ_i ← argmax_{1≤θ≤γ_i} {∀ j: 1 ≤ j ≤ n − θ:
                t_{i+1,j} ≥ τ_{i,j+θ} + (θ+1)t̃};
  if (θ_i > 0) then
    for h = 1 to θ_i do
      Apply Equations (17)–(19) to insert
        an emergency train into X';
      p ← p − 1;
      if (p = 0) then return X' end if;
        //scheduling succeeds
    end for
  end if
end for
return X'; //scheduling fails.
```

4.1.2. Ideal Solutions with Overtaking.

How can an emergency train e "ideally" overtake its preceding train i stopping at station s? The premise is that the dwell time of i at station s should be sufficient for e to pass through sections $s-1$ and s

$$\Delta t_{i,s} \geq d_{s-1}/v^{\mathrm{U}}_{s-1} + d_s/v^{\mathrm{U}}_s + 2\tilde{t}. \tag{20}$$

Let S^*_i denote the set of stations satisfying Equation (20). For convenience, we define two predicates $Ideal(i,s,s')$ and $Ideal^{\mathrm{NO}}(i,s,s')$ to denote whether an emergency train can be ideally inserted immediately after i from station s to station s' when overtaking *is* and *is not* allowed, respectively. We already know the latter can be evaluated as

$$Ideal^{\mathrm{NO}}(i,s,s') \equiv (\Theta(i,s,s') > 0). \tag{21}$$

The former can be recursively evaluated as

$$Ideal(i,s,s') \equiv \begin{cases} \text{true} \quad (i=0) \vee Ideal^{\mathrm{NO}}(i,s,s') \\ (\exists s^*\colon s^* \in S^*_i \wedge s \leq s^* < s'\colon Ideal^{\mathrm{NO}}(i,s,s^*) \\ \quad \wedge\, Ideal(i-1,s^*,s')) \quad \text{otherwise.} \end{cases} \tag{22}$$

If $Ideal(i,1,n+1)$ is false for any $1 \leq i < m$, no emergency train can be inserted. Otherwise, let i^* be the smallest index satisfying that $Ideal(i^*,1,n+1)$, the earliest emergency train (the uppermost emergency flow line) that can be ideally inserted into X is scheduled as follows (the pseudo-code is given by Algorithm B1 in Online Appendix B):

(1) Traveling immediately after i^* by applying Equations (17)–(19) unless an overtaking is feasible.

(2) Overtaking the preceding train(s) whenever overtaking is feasible, and set i^* to the new preceding train it does not overtake.

(3) If $i^* = 0$, traveling with the maximum velocity to the last station.

(4) Repeating steps (1)–(3) until reaching the last station.

Theorem 2. *Let e^* be the earliest emergency train that can be ideally inserted into X, then e^* can always be included in an ideal solution.*

Let $\Lambda(X)$ be the maximum number of emergency trains that can be ideally inserted into timetable X, and $X\downarrow_e$ be the timetable produced by inserting an emergency train e into X, then Theorem 2 indicates

$$\Lambda(X) = 1 + \Lambda(X\downarrow_{e^*}). \tag{23}$$

The principle of optimality is, in fact, the foundation of Algorithms 1 and B1, and its proof is trivial when overtaking is not allowed. The proof of Theorem 2, when overtaking is allowed, is given in Online Appendix B. Based on Equation (23), we develop Algorithm 2 for finding an ideal solution to the problem when overtaking is allowed.

Algorithm 2 (Searching for an ideal solution with overtaking)

Input A predefined timetable $X = (I, J, S, d[n], v^{\mathrm{U}}[m,n], t[m+1,n], \hat{t}[m+1,K], \tilde{t})$, the number p of the emergency trains, and the maximum velocity vector $v^{\mathrm{U}}[n]$

Output An ideal timetable X'

let $X' = X, i = 1$; *//i is used to keep the position before which the insertion has been done*
while $i \leq m$ **do**
  let $i^* = \arg\min_{i \leq i' \leq m}\{Ideal(i',1,n+1)\}$;
  if $(i^* = m)$ **then break**; *//scheduling fails*
  Create an earliest emergency schedule e^* according to Steps (1)–(4) in Section 4.1.2;
  $X' \leftarrow X'\downarrow_{e^*}$, $i \leftarrow i^*+1$, $p \leftarrow p-1$;
  if $(p = 0)$ **then return** X'; *//scheduling succeeds*
end while
return X'.

The nonrecursive procedure for evaluating $Ideal(i,1,n+1)$ is given in Online Appendix B, which has a time complexity of $O(i \cdot n)$. Thus Algorithm 2 has a worst-case time complexity of $O((m(m+p)/2)n)$. Yet in most cases the efficiency of Algorithm 2 is close to Algorithm 1, because the size of S^*_i is much smaller than n and the number of selected i^* is much smaller than m.

4.2. Quasi-Ideal Solutions

4.2.1. Lifting and Lowering.

A quasi-ideal solution allows regular trains to change their schedules in some intermediate sections, as long as it will not delay their arrival at any planned stop. Of course, such changes should facilitate the insertion of emergency trains. From the perspective of the time-space network, the changes can be divided into two types:

- *Lifting* the flow line (or a part of it) of a regular train, i.e., speeding up the train to increase the headroom between the train and its succeeding train.
- *Lowering* the flow line (or a part of it) of a regular train, i.e., slowing down the train to increase the headroom between the train and its preceding train.

4.2.2. Quasi-Ideal Solutions Without Overtaking.

Without overtaking, the lifting or lowering should be done for increasing the headroom over the whole line rather than part of it.

The first train should be lifted as much as possible while satisfying constraint (7), thus we schedule it as

$$t^{\mathrm{R}}_{1,1} = t_{1,1}, \tag{24}$$

$$t^{\mathrm{R}}_{1,j+1} = \begin{cases} t_{1,j+1} & (j+1) \in S_1 \\ t^{\mathrm{R}}_{1,j} + d_j/v^{\mathrm{U}}_{1,j} & (j+1) \notin S_1, \end{cases} \tag{25}$$

$$\hat{t}^{\mathrm{R}}_{1,j+1} = t^{\mathrm{R}}_{1,j} + d_j/v^{\mathrm{U}}_{1,j}, \quad (j+1) \in S_1. \tag{26}$$

Each intermediate train i $(1 < i \leq m)$ may be either lifted or lowered. Suppose the schedule of the preceding train $i-1$ is fixed, we can draw an *upper limit flow* *line* and a *lower limit flow line* of i, denoted as ULFL(i) and LLFL(i), respectively, to limit the range of lifting and lowering:

• ULFL(i) advances i as much as possible but still keeps the minimum safety distance to $i-1$ and does not violate constraint (4), and thus its time points (marked with a superscript U) are determined as

$$t^{\mathrm{U}}_{i,j+1}=\begin{cases} t_{i,j+1} & (j+1)\in S_i \\ \max(\tau^{\mathrm{R}}_{i-1,j+1}+\tilde{t},t^{\mathrm{U}}_{i,j}+d_j/v^{\mathrm{U}}_{i,j}) & (j+1)\notin S_i, \end{cases} \tag{27}$$

$$\hat{t}^{\mathrm{U}}_{i,j+1}=t^{\mathrm{U}}_{i,j}+d_j/v^{\mathrm{U}}_{i,j},\quad (j+1)\in S_i. \tag{28}$$

• LLFL(i) delays i as much as possible but still enables it to arrive at each planned stop on time, and thus its time points (marked with a superscript L) are determined (in a backward manner) as

$$\hat{t}^{\mathrm{L}}_{i,j}=\hat{t}_{i,j},\quad j\in S_i, \tag{29}$$

$$t^{\mathrm{L}}_{i,j}=\begin{cases} \hat{t}_{i,j+1}-d_j/v^{\mathrm{U}}_{i,j} & (j+1)\in S_i \\ t^{\mathrm{L}}_{i,j+1}-d_j/v^{\mathrm{U}}_{i,j} & (j+1)\notin S_i. \end{cases} \tag{30}$$

There can be seemingly infinite alternatives between ULFL and LLFL. However, those alternatives can be partitioned into a finite number of equivalence classes. Let θ^{L}_i be the headroom between $i-1$ and ULFL(i) and θ^{U}_i be the headroom between $i-1$ and LLFL(i). We can use Equations (31) and (32) to determine $(\theta^{\mathrm{U}}_i-\theta^{\mathrm{L}}_i+1)$ alternatives, each of which keeps an exact headroom of θ between $i-1$ and i while leaving as much headroom after i as possible $(\theta=\theta^{\mathrm{L}}_i,\theta^{\mathrm{L}}_i+1,\ldots,\theta^{\mathrm{U}}_i)$

$$t^{\mathrm{R}}_{i,j}=\begin{cases} \max(t_{i,j},\tau^{\mathrm{R}}_{i-1,j}+\theta d_j/v^{\mathrm{U}}_j+(\theta+1)\tilde{t}) \\ \qquad j\in S_i\wedge j>n-\theta \\ \max(t_{i,j},\tau^{\mathrm{R}}_{i-1,j}+\theta d_j/v^{\mathrm{U}}_j+(\theta+1)\tilde{t}, \\ \qquad \tau^{\mathrm{R}}_{i-1,j+\theta}+(\theta+1)\tilde{t}) \quad j\in S_i\wedge j\le n-\theta \\ \max(t^{\mathrm{R}}_{i,j-1}+d_{j-1}/v^{\mathrm{U}}_{i,j-1},\tau^{\mathrm{R}}_{i-1,j}+\theta d_j/v^{\mathrm{U}}_j \\ \qquad +(\theta+1)\tilde{t}) \quad j\notin S_i\wedge j>n-\theta \\ \max(t^{\mathrm{R}}_{i,j-1}+d_{j-1}/v^{\mathrm{U}}_{i,j-1},\tau^{\mathrm{R}}_{i-1,j}+\theta d_j/v^{\mathrm{U}}_j+(\theta+1)\tilde{t}, \\ \qquad \tau^{\mathrm{R}}_{i-1,j+\theta}+(\theta+1)\tilde{t}) \quad j\notin S_i\wedge j\le n-\theta, \end{cases} \tag{31}$$

$$\hat{t}^{\mathrm{R}}_{i,j+1}=t^{\mathrm{R}}_{i,j}+d_j/v^{\mathrm{U}}_{i,j},\quad (j+1)\in S_i. \tag{32}$$

Let Φ_i be the set of these $(\theta^{\mathrm{U}}_i-\theta^{\mathrm{L}}_i+1)$ flow lines, and for each $l_i\in\Phi_i$ let $h(l_i)$ be the headroom between $i-1$ and l_i. Any other alternative can be replaced by an $l_i\in\Phi_i$, which has the same headroom between $i-1$ and i but can leave more (or at least the same) time slots after i, as illustrated in Figure 3 (where the two other alternative flow lines can be replaced by the ULFL).

The set Φ_i has $(\theta_i-\theta^{\mathrm{L}}_i+1)$ *lifted* flow lines and $(\theta^{\mathrm{U}}_i-\theta_i)$ *lowered* flow lines. Note that, for the lowered flow lines, we do not consider the minimum safety distance between i and $i+1$, because the subsequent regular trains can also be lowered. The remaining time period in each section j should be at least sufficient for passing the remaining regular trains

$$\tau_{i,j}+(m-i)\tilde{t}\le \dot{t}_j-\sum_{i'=i+1}^{m}(d_j/v^{\mathrm{U}}_{i',j}),\quad 1\le j\le n. \tag{33}$$

Condition (33) can be enhanced by an upper bound limit p_{best}, which is the number of insertions of a best known solution: if the current partial solution has successfully inserted p_c emergency trains before i, the number of new trains that can be further inserted after i is at most

$$p_r=\min_{1\le j\le n}\left\{\left(\dot{t}_j-\tau_{i,j}-\sum_{i'=i+1}^{m}(d_j/v^{\mathrm{U}}_{i',j})-(m-i)\tilde{t}\right)\frac{1}{d_j/v^{\mathrm{U}}_j+\tilde{t}}\right\}, \tag{34}$$

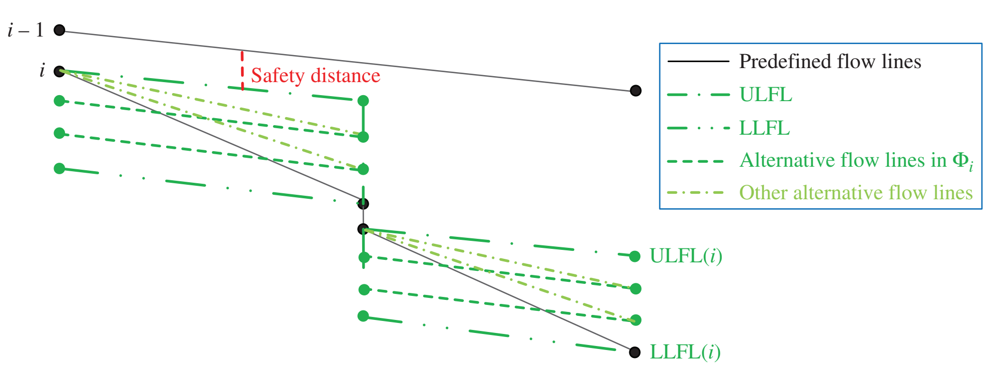
so if $p_c+p_r\le p_{\mathrm{best}}$, the current solution does not need to be considered further.

Let $X\uparrow_{l_i}$ be the timetable produced by changing the schedule of train i to $l_i\in\Phi_i$, and $\Lambda(X,i)$ be the maximum number of emergency trains that can be "quasi-ideally" inserted after i in X, then the number can be recursively calculated as

$$\Lambda(X,i-1)=\begin{cases} 0 & i>m \\ \max_{l_i\in\Phi_i}\{h(l_i)+\Lambda(X\uparrow_{l_i},i)\} & 1<i\le m. \end{cases} \tag{35}$$

Such a recursive procedure can be easily transformed to a nonrecursive one as shown in Algorithm 3, where

Figure 3. (Color online) Illustration of ULFL, LLFL, and Alternative Flow Lines



$\Lambda(X_c, i_c, p_c)$ denotes a search node with a timetable X_c produced by inserting p_c emergency trains before i_c.

Algorithm 3 has a worst-case time complexity of $O(np^m)$ and an average time complexity of the order $O(n(\overline{\theta^{\mathrm{U}} - \theta^{\mathrm{L}} + 1})^m)$, where $(\overline{\theta^{\mathrm{U}} - \theta^{\mathrm{L}} + 1})$ denotes the average value of $(\theta_i^{\mathrm{U}} - \theta_i^{\mathrm{L}} + 1)$ among the regular trains. For a compact HSR timetable, most $(\theta_i^{\mathrm{U}} - \theta_i^{\mathrm{L}} + 1)$ values are small (and are even 1 for trains during peak hours); this fact, in conjunction with the pruning procedure, makes the algorithm efficient in solving most HSR emergency scheduling problem instances.

Algorithm 3 (Searching for a quasi-ideal solution without overtaking)

Input A predefined timetable $X = (I, J, S, d[n], v^{\mathrm{U}}[m,n], t[m,n], \hat{t}[m,K], \dot{t}[n], \tilde{t})$, the number p of the emergency trains, and the maximum velocity vector $v^{\mathrm{U}}[n]$
Output A quasi-ideal timetable X'

1 **call** Algorithm 1 to get the maximum number of nonovertaking trains of an ideal solution as p_{best};
2 Initialize a stack Q with a single node $\Lambda(X, 1, 0)$;
3 **while** Q is not empty **do**
4 Take the first node $\Lambda(X_c, i_c, p_c)$ from Q;
5 Compute the set Φ_{i_c} of alternative flow lines for i_c in X_c according to Equations (31) and (32);
6 **for each** $l_i \in \Phi_{i_c}$ **do**
7 **let** $p_n = \min(p - p_c, h(l_i))$;
8 Create a new timetable X_c' by inserting p_n emergency trains between $(i_c - 1)$ and i_c in X_c;
9 $p_c \leftarrow p_c + p_n$;
10 **if** $(p_c \geq p)$ **then return** X_c'; *//scheduling succeeds*
11 Compute p_r according to Equation (34);
12 **if** $(p_c + p_r > p_{\mathrm{best}})$ **then**
13 Add $\Lambda(X_c', i_c + 1, p_c)$ to Q;
14 $p_{\mathrm{best}} \leftarrow \max(p_{\mathrm{best}}, p_c)$; *//updating the bound*
15 **end if**
16 **end for**
17 **end while**
18 **return** the timetable of the node with p_{best}; *//scheduling fails.*

4.2.3. Quasi-Ideal Solutions with Overtaking. When overtaking is allowed, expanding headroom over the sections between two intermediate stations (rather than over the whole line) may also contribute to the increase in the number of emergency trains, because an emergency train may overtake a regular train at an intermediate station. In addition to Equations (21) and (22), we define the following predicate to denote whether an emergency train can overtake train i at station s and then travel to s':

$$Ideal^{\mathrm{OT}}(i, s, s') \equiv (s \in S_i^*) \wedge ((i = 0) \vee (Ideal(i-1, s, s'))). \quad (36)$$

Let S_i^{OT} be the set of stations that satisfies $Ideal^{\mathrm{OT}}(i, s, n+1)$ and assume the stations are ordered by their numbers. For each pair of adjacent stations $s, s' \in S_i^{\mathrm{OT}}$, we compute ULFL and LLFL of i in the range of $[s, s']$ (the headroom between $i-1$ and the two flow lines are denoted by $\theta_{i,s}^{\mathrm{L}}$ and $\theta_{i,s}^{\mathrm{U}}$, respectively), and then determine $(\theta_{i,s}^{\mathrm{U}} - \theta_{i,s}^{\mathrm{L}} + 1)$ alternative subschedules using Equations (31) and (32) (where the range of j is limited to $[s, s']$).

Between trains $i-1$ and i, in addition to the $(\theta_i^{\mathrm{U}} - \theta_i^{\mathrm{L}} + 1)$ flow lines in Φ_i, we can further draw a set Ψ_i of flow lines by connecting those alternative subschedules. However, the connection of two adjacent subschedules at station s is feasible only when the interval between the arrival time $\hat{t}_{i,s}$ of the first subschedule and the leaving time $t_{i,s}$ of the second subschedule is sufficient for an emergency train to overtake, i.e., it satisfies Equation (20).

Now we have at most $|\Phi_i \cup \Psi_i|$ possible flow lines for each i when overtaking is allowed $(1 < i \leq m)$. For each $l_i \in \Psi_i$ let $h'(l_i)$ be the number of emergency trains that can be inserted between $i-1$ and l_i, then Equation (35) becomes

$$\Lambda(X, i) = \begin{cases} 0 & i > m \\ \max\left(\max\limits_{l_i \in \Phi_i}\{h(l_i) + \Lambda(X \uparrow_{l_i}, i+1)\},\right. \\ \qquad \left.\max\limits_{l_i \in \Psi_i}\{h'(l_i) + \Lambda(X \uparrow_{l_i}, i+1)\}\right) \\ \quad 1 \leq i \leq m. \end{cases} \quad (37)$$

Thus we get Algorithm 4, whose pruning procedure uses the total remaining headroom in Section 1 as the maximum possible number of emergency trains that can be further inserted (Line 12).

Algorithm 4 (Searching for a quasi-ideal solution with overtaking)

Input A predefined timetable $X = (I, J, S, d[n], v^{\mathrm{U}}[m,n], t[m,n], \hat{t}[m,K], \dot{t}[n], \tilde{t})$, the number p of the emergency trains, and the maximum velocity vector $v^{\mathrm{U}}[n]$
Output A quasi-ideal timetable X'

1 **call** Algorithm 2 to get the maximum number of trains of an ideal solution as p_{best};
2 Initialize a stack Q with a single node $\Lambda(X, 1, 0)$;
3 **while** Q is not empty **do**
4 Take the first node $\Lambda(X_c, i_c, p_c)$ from Q;
5 Compute the sets Φ_{i_c} and Ψ_{i_c} of alternative flow lines for i_c in X_c;
6 Perform Lines 6–16 of Algorithm 3;
7 **for each** $l_i \in \Psi_{i_c}$ **do**
8 **if** l_i does not satisfy Equation (33) **then continue**; *//infeasible*
9 **let** $p_n = \min(p - p_c, h'(l_i))$;
10 Create a new timetable X_c' by inserting p_n emergency trains between $(i_c - 1)$ and i_c in X_c;
11 $p_c \leftarrow p_c + p_n$;
12 **if** $(p_c \geq p)$ **then return** X_c'; *//scheduling successes*

```
        if (p_c + Σ_{i=i_c}^{m} Θ(i,1,2) > p_best) then
            add Λ(X'_c, i_c + 1, p_c) to Q;
            p_best ← max(p_best, p_c); //updating the bound
        end if
    end for
end while
return the timetable of the node with p_best;
    //scheduling fails.
```

4.3. Feasible Solutions

In comparison with a quasi-ideal solution, a feasible solution allows each regular train i to be delayed at any station, as long as it does not decrease any planned dwell time and can arrive at the last station before the deadline $\ddot{t}_i$. When searching for a feasible solution, ULFL(i) is also calculated by Equations (27) and (28), but LLFL(i) is calculated as follows, which may be more relaxed than Equations (29) and (30):

$$\hat{t}^{\mathrm{L}}_{i,n+1} = \ddot{t}_i, \tag{38}$$

$$t^{\mathrm{L}}_{i,j} = \begin{cases} \hat{t}^{\mathrm{L}}_{i,j+1} - d_j/v^{\mathrm{U}}_{i,j} & (j+1) \in S_i \\ t^{\mathrm{L}}_{i,j+1} - d_j/v^{\mathrm{U}}_{i,j} & (j+1) \notin S_i, \end{cases} \tag{39}$$

$$\hat{t}^{\mathrm{L}}_{i,j} = t^{\mathrm{L}}_{i,j} - \Delta t_{i,j} \quad j \in S_i. \tag{40}$$

By replacing Equations (29) and (30) with Equations (38)–(40) for calculating LLFLs, Algorithms 3 and 4 can also be applied to find feasible solutions without and with overtaking, respectively.

Remark 5. As the objective of our problem is defined as to minimize the total arrival time of the emergency trains, the rescheduling space of regular trains might be large, thus there might be many optimal solutions, but among them the solution produced by our method has the minimum total delay of regular trains, because it always uses exact, nonredundant headroom to insert emergency trains by Equations (31) and (32).

4.4. Removing the Two Temporary Assumptions

When the two temporary assumptions at the end of Section 3.1 do not hold, we perform preprocessing to convert the original regular timetable to a "virtual" timetable satisfying the assumptions.

Å1. There is no overtaking among the regular trains in the original timetable.

When overtaking occurs among the regular trains in the original timetable, for all parts of the flow lines between each pair of adjacent stations s and $s+1$, we always assign the uppermost one to regular train 1, assign the second uppermost one to train 2, ..., and finally assign the lowermost one to the last train m. However, lifting and lowering of flow lines must be done before renumbering, to avoid miscalculation of headroom between alternative flow lines because of the change of predefined dwell time.

Å2. All regular trains travel from the first station to the last station.

If a train i starts from the first station but terminates at an intermediate station s, when calculating the headroom between $i-1$ and i, we assume that i virtually shares the part of flow line of $i+1$ after s (but the safety constraint between i and $i+1$ after s is neglected; if i is the last train, its flow line connects the latest time points $\dot{t}_j$ after s); when calculating the headroom between i and $i+1$, we assume that i virtually shares the part of flow line of $i-1$ after s (but the safety constraint between $i-1$ and i after s is neglected; if i is the first train, its flow line overlaps the horizontal axis after s).

Similarly, if a train i starts from an intermediate station s and then travels to the last station, when calculating the headroom between $i-1$ and i, we assume that i virtually shares the part of flow line of $i+1$ before s; when calculating the headroom between i and $i+1$, we assume that i virtually shares the part of flow line of $i-1$ before s.

If a train i starts from an intermediate station s and terminates at another intermediate station s', we virtually extend its flow line to the whole HSR line by combining the above two methods.

5. Numerical Experiments

The proposed solution method has been applied to evaluate the capacity of emergency train scheduling on 23 HSR lines, which cover almost all HSR lines that were operational by the end of 2014 in China. In this section we present numerical results on eight typical lines, including four bullet-train special lines and four lines carrying both bullet and traditional trains. Table 1 summarizes the main features of the eight lines. The experimental environment is a computer with an Intel i7-4500 CPU, 8 GB DDR3 memory, and a Windows 7 system. The algorithms are implemented by Visual C# 2012. The experiments are divided into two groups.

Table 1. The Summary of the Eight HSR Timetables Used in the Experiments

Line	Length	V_D	K	n	m	m^{O}	V_R^{U}	V_R^{L}	Hybrid
Chang–Jiu	135	250	6	27	35	12	223	139	Y
Jing–Jin	165	350	7	32	88	0	306	267	N
Ning–Hang	249	350	11	42	51	0	303	192	N
Donghuan	308	200	14	63	28	2	176	136	Y
He–Wu	359	250	14	78	53	6	225	137	Y
Hu–Chang	1,062	350	29	172	49	0	298	206	N
Hang–Shen	1,495	250	65	299	58	8	221	156	Y
Jing-Guang	2,298	350	36	391	75	0	302	197	N

Notes. V_D, The line's design speed (km/hr); K, number of stations; n, number of sections; m, number of regular trains; m^{O}, number of traditional trains in the timetable; V_R^{U}, maximum average velocity of regular trains (km/hr); V_R^{L}, minimum average velocity of regular trains (km/hr); Hybrid, carrying both bullet and traditional trains.

5.1. Algorithm Performance on Given Sets of Emergency Trains

In the first group, on each of the eight HSR lines, we construct 19 problem instances, which consider inserting $10, 15, 20, \ldots, 100$ emergency trains into the regular HSR timetable, respectively. The maximum allowable velocity v_j^{U} of the emergency trains is always set to $0.95V_D$ (V_D is the design speed of the HSR line), and the buffer time $\tilde{t}$ is set to 60 seconds when $V_D \le 300$ and 90 seconds when $V_D > 300$. For Q3, the deadline $\ddot{t}_i$ is always set to $t_{i,n+1} + 0.1(t_{i,n+1} - t_{i,1})$, i.e., the maximum delay is one-tenth the train's total travel time. We, respectively, run Algorithm 2, Algorithm 4, and its variant to search the ideal, quasi-ideal, and feasible solutions of each problem instance, where overtaking is always allowed. For comparison, we also run CPLEX 12.6 to solve the problems as integer programs (with the automatic cut generator switched on and priority given to the variables related to emergency trains). The experimental results show that on each instance, our approach and CPLEX always obtain the same objective value, which demonstrates that our approach guarantees the optimal solution as the basic integer programming algorithm.

Figure 4(a)–4(h) compares the CPU time of the algorithms for solving the instances. Using our method, most of the instances are solved within one second, except that finding feasible solutions on the last two lines needs less than 10 and 20 seconds, respectively. However, CPLEX consumes much more time, which is about 6~60 on small instances (typically with $p \le 60$) and exceeds one hour on large instances (when $p \ge 80$ on the last three lines). Thus our method is more suitable than the CPLEX solver for train scheduling in emergency situations.

For our method, the increase of time with the number p of emergency trains is slow for finding ideal solutions, but relatively fast for quasi-ideal solutions, and very fast for feasible solutions (the tails of some curves become flat because p exceeds the maximum allowable number of emergency trains). This is because the time complexity of Algorithm 2 is polynomial but that of Algorithm 4 is exponential. Nevertheless, the increase of CPU time in Algorithm 4 with instance size is not significant, mainly because of its effective pruning procedure.

Moreover, the CPU time of our method for ideal solutions is not affected much by the velocity of regular trains. Yet the lower the velocity of the regular trains, the larger the (average) headroom between them, and the more time required for finding feasible solutions when p is not small, e.g., the instance sizes on the Ning–Hang line (with high regular velocity) and the Donghuan line (with low regular velocity) are similar, but T_f on the latter is much larger than that on the former.

We also observe that the CPU time for quasi-ideal and feasible solutions on a hybrid line is usually more than that on a special line. To further validate this point, we temporarily replace about half of the traditional trains with high-speed trains in the Chang–Jiu and Donghuan timetables and then retest the solution time. The results show that the time decreases around 8%~30% on the new timetables. This is because a hybrid timetable with more lower-speed trains often involves more overtakings, and thus Algorithm 4 has to check a larger number of alternative flow lines.

The CPU time consumed on an instance represents the difficulty of solving the instance. Thus the experiments reveal that, besides the instance size, the low velocity of regular trains and the hybridization of bullet and traditional trains also increase the difficulty of an instance.

5.2. Maximum Number of Emergency Trains

In the second group, on each of the eight HSR lines, we construct $11 \times 13 = 143$ problem instances, which consider determining the maximum number of emergency trains that can be inserted into the regular HSR table when v_j^{U} is set to $V_D, 0.99V_D, \ldots, 0.9V_D$ and $\tilde{t}$ is set to $\tilde{t}_0, \tilde{t}_0 + 5, \ldots, \tilde{t}_0 + 60$, respectively, where $\tilde{t}_0$ (in seconds) is determined as follows:

$$\tilde{t}_0 = \begin{cases} 45 & v_j^{\mathrm{U}} \le 200 \\ 60 & 200 < v_j^{\mathrm{U}} \le 300 \\ 90 & v_j^{\mathrm{U}} > 300. \end{cases} \tag{41}$$

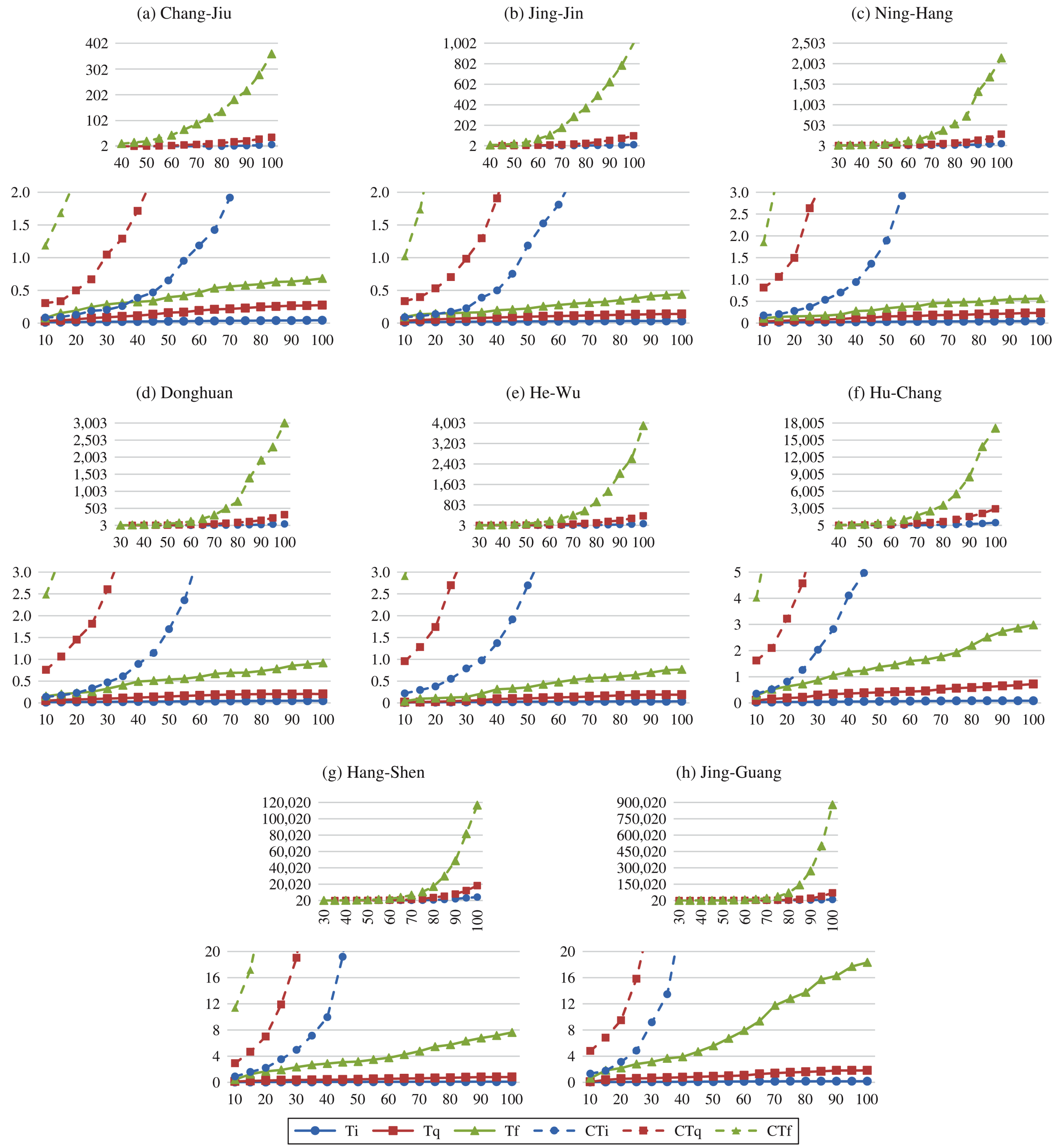
Remark 6. Generally, $\tilde{t}$ should be set proportional to the maximum allowable velocity. Equation (41) is empirical, and we suggest that $\tilde{t}$ should not be less than $\tilde{t}_0$ for keeping the schedules dependable and robust. A deep analysis on the sensitivity to the buffer time is part of our future work.

We, respectively, run Algorithm 4 and its variant to search the quasi-ideal and feasible solutions of each problem instance, where overtaking is always allowed. For feasible solutions, the maximum delay of each train is one-tenth the train's total travel time.

Figure 5 presents the maximum number p_{best} of emergency trains of the quasi-ideal and feasible solutions on the eight HSR lines. The Chang–Jiu line has the maximum capacity for inserting emergency trains because its number of regular trains is the smallest among the eight lines. The He–Wu line has the minimum capacity. Note that the number of regular trains on the He–Wu line is smaller than the Hang–Shen and Jing–Guang lines, but the latter two lines are longer and have more trains that start from and/or terminate at intermediate stations.

We are more interested in how p_{best} changes with the train velocity and the buffer time. Obviously, p_{best}

Figure 4. (Color online) The CPU Time Consumed for Finding Three Kinds of Solutions on the Eight HSR Lines

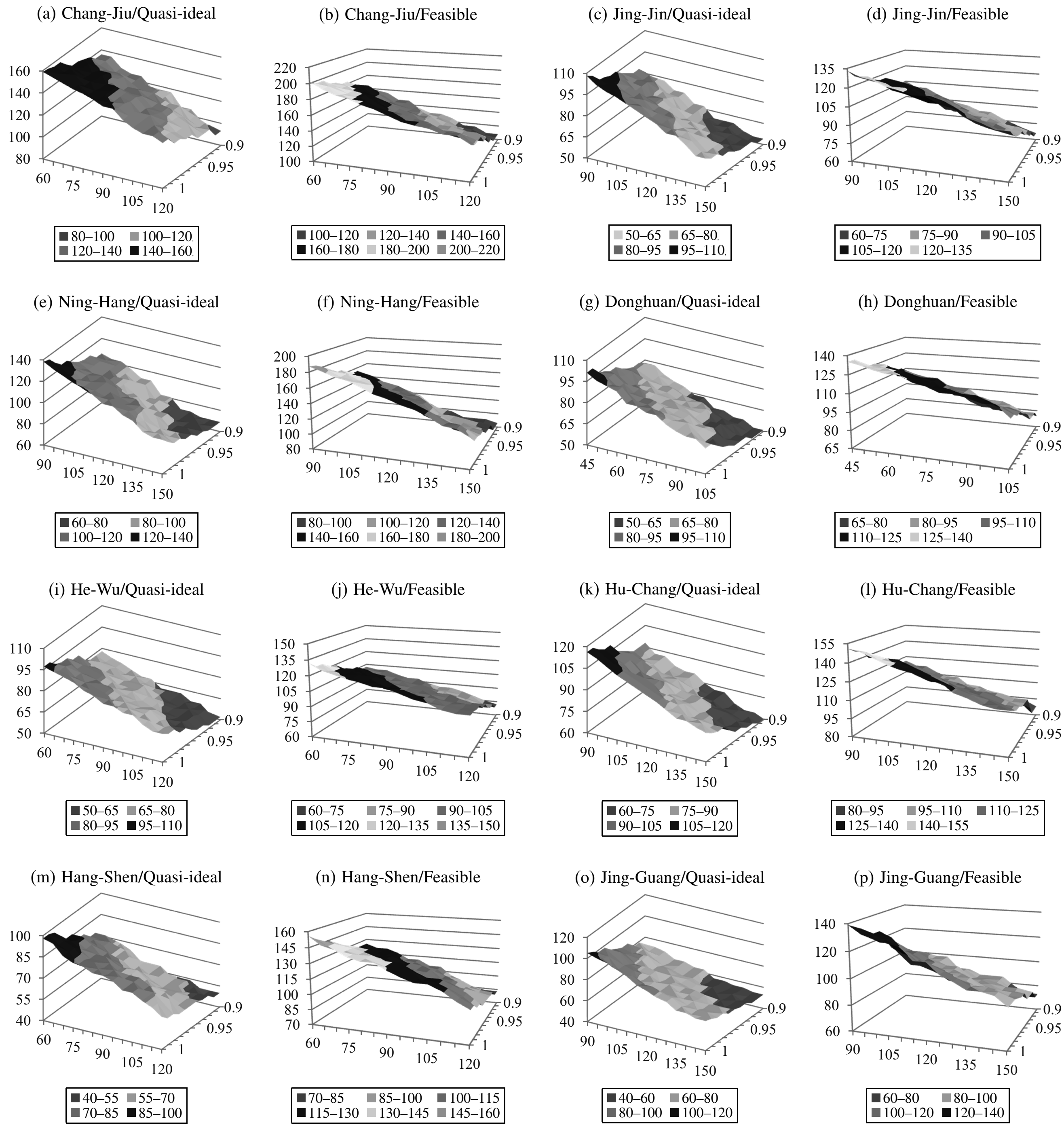


Notes. The horizontal axis denotes the number p of emergency trains, and the vertical axis denotes the CPU time (in seconds). Ti, Tq, and Tf denote the time of the proposed algorithms for finding ideal, quasi-ideal, and feasible solutions, respectively; CTi, CTq, and CTf denote the corresponding solution time of CPLEX. In each subfigure, the upper-right view presents the complete curves of the CPLEX solver (whose time scale will make the curves of our method illegible).

decreases with the decrease of v_j^{U} or with the increase of $\tilde{t}$. It can be observed from Figure 5:

- On each of the 3D surfaces, the curves on the front side are steeper than the curves on the back side. This indicates that the decline of p_{best} over $\tilde{t}$ when the velocity v_j^{U} is fixed to a higher value is faster than that when v_j^{U} is fixed to a lower value.
- On each of the 3D surfaces, the curves on the left side are steeper than the curves on the right side. This indicates that the decline of p_{best} over v_j^{U} when $\tilde{t}$ is fixed

Figure 5. (Color online) The Maximum Number of Emergency Trains of Quasi-Ideal and Feasible Solutions on the Eight HSR Lines



Note. The x-axis denotes the buffer time, the y-axis denotes the rate of v_j^{U} to V_D, and the z-axis denotes the p_{best} value.

to a smaller value is faster than that when v_j^{U} is fixed to a larger value.

That is, the higher the velocity v_j^{U} (or the smaller the buffer time $\tilde{t}$), the higher the sensitivity of p_{best} to $\tilde{t}$ (or v_j^{U}). Moreover, on each HSR line, the curves of the feasible solutions are steeper than their counterparts of the quasi-ideal solutions, which indicates that p_{best} is more sensitive to v_j^{U} or $\tilde{t}$ when the constraints are more severe.

6. A Real-World Application

On August 12, 2015, a chemical explosion occurred at a container storage station at the Port of Tianjin, North China. Fires caused by the initial explosions continued to burn uncontrolled throughout the night, repeatedly causing secondary explosions. As of September 12, 2015, the official casualty report was 173 deaths, 8 missing, and 797 nonfatal injuries (Tianjin Daily 2015).

The rescue headquarters planned 36 emergency trains to deliver relief personnel, equipment, and supplies to the affected area through the Jing–Jin HSR line, the regular timetable of which was summarized in the third row of Table 1—the line is not long, but the train speed and frequency are high. Among the 36 emergency trains, 10 were to be dispatched on the afternoon of August 13 and 26 were to be dispatched on August 14. The headquarters also observed a significant increase in passenger volume (the number of tickets booked) in the two days after the disaster. So it was expected to produce quasi-ideal solutions, which would not delay the regular passengers, to not increase public dissatisfaction.

Although we had worked out a set of emergency schedules on the line in advance (see Figure 5(c) and 5(d)), the schedules were in terms of whole days. Therefore, for the first set of 10 emergency trains that were to be scheduled after 12:00, we constructed a new problem instance where $v_j^{\mathrm{U}} = 320$ km/hr and $\tilde{t} = 105$ seconds, and used Algorithm 4 to produce a solution summarized in columns (2)–(5) of the second row of Table 2. The algorithm running time was less than one second, and the time interval from the receipt of the requirement to the submission of the solution was less than two minutes. The implementation result of the solution was summarized in columns (2)–(5) of the third row of Table 2. The average real travel time (35.27 minutes) of the emergency trains was a little (2.92 minutes) more than that in the schedule; the schedule was a quasi-ideal solution, but its real implementation was somewhat deviated and caused eight regular trains arriving later in some planned stops (6 minutes on average).

After analyzing the implementation of the first solution, the headquarters concluded that it was of high quality, but the railway department asked whether the delay of the regular trains could be further suppressed. Therefore, for the second set of 26 trains, we selected the quasi-ideal solution of the instance where v_j^{U} is still 320 km/hr but $\tilde{t}$ is 120 seconds from the set of precalculated solutions on this HSR line. The solution and its real implementation were, respectively, summarized in columns (6)–(9) of the second and third rows of Table 2. The average scheduled travel time of this set was about 2.5 minutes more than that of the first set, not only because the buffer time was increased but also because in the first set the emergency trains departing in the afternoon had more chances to overtake regular trains. In comparison, in the second set, the average increment of the real travel time over the scheduled time was only 1.95 minutes, which was much less than the first set; the 10 emergency trains in the first set caused a delay of eight regular trains, while the 26 emergency trains in the second set only caused a delay of nine regular trains, whose average delay time was only 3.87 minutes. This showed that our adjustment of $\tilde{t}$ decreased the sensitivity and increased the effectiveness of the emergency schedules.

In summary, our solution method was successfully applied to the real-world emergency operation: the solution time was very short, and the produced solutions were relatively accurate and reliable.

Table 2. The Summary of the Two Solutions and Their Implementations for Emergency Train Scheduling in Rescue Operations for the 2015 Tianjin Chemical Explosion

	August 13				August 14			
	$\bar{t}$	N^{OT}	N^{D}	t^{D}	$\bar{t}$	N^{OT}	N^{D}	t^{D}
Solution	32.35	6.70	0	0	34.80	3.31	0	0
Implementation	35.27	6.50	8	6.33	36.75	3.12	9	3.87

Notes. $\bar{t}$, The average travel time (in minutes) of the emergency trains; N^{OT}, the average number of regular trains overtaken by the emergency trains; N^{D}, the number of delayed regular trains; t^{D}, the average delay time (in minutes) of the delayed regular trains.

7. Conclusion

We consider a problem of scheduling emergency trains within a predefined HSR timetable, and classify solutions to the problem into ideal, quasi-ideal, and feasible solutions. An ideal solution is easy to produce using our definition of the headroom function, and it lays the foundation for finding quasi-ideal and feasible solutions. Numerical experiments and the application to the 2015 Tianjin chemical explosion demonstrate the effectiveness and efficiency of our solution method.

Besides directly producing emergency schedules, our method also enables the evaluation of the emergency capacity of HSR lines, which is important in the decision of new HSR line construction and timetabling. China plans to extend the length of its HSR network to more than 40,000 km by 2025 (NDRC 2016); moreover it takes HSR as a key part of the "One Belt and One Road" strategy, and has established HSR cooperation with over 50 countries and regions in the world. Thus our method is expected to be used in a much wider context in the future.

The current problem concerns the emergency trains traveling through a single line. We are currently extending it to more complex HSR networks and using hierarchical optimization and metaheuristics (Zheng 2015, Zheng, Chen, and Ling 2015a) to solve the extended problems.

References

Adenso-Diaz B, Gonzalez MO, Gonzalez-Torre P (1999) On-line timetable re-scheduling in regional train services. *Transportation Res. Part B* 33:387–398.

Albrecht AR, Panton DM, Lee DH (2013) Rescheduling rail networks with maintenance disruptions using problem space search. *Comput. Oper. Res.* 40:703–712.

Brannlund U, Lindberg PO, Nou A, Nillson J-E (1998) Railway timetabling using Lagrangian relaxation. *Transportation Sci.* 32(4):358–369.

Burdett RL, Kozan E (2009) Techniques for inserting additional trains into existing timetables. *Transportation Res. Part B* 43:821–836.

Burdett RL, Kozan E (2010) A sequencing approach for creating new train timetables. *OR Spectrum* 32:163–193.

Cacchiani V, Caprara A, Toth P (2010) Scheduling extra freight trains on railway networks. *Transportation Res. Part B* 44:215–231.

Cai X, Goh CJ, Mees AI (1998) Greedy heuristics for rapid scheduling of trains on a single track. *IIE Trans.* 30:481–493.

Cao J, Liu XC, Wang Y, Li Q (2013) Accessibility impacts of China's high-speed rail network. *J. Transportation Geography* 28:12–21.

Caprara A, Fischetti M, Toth P (2002) Modeling and solving the train timetabling problem. *Oper. Res.* 50(4):851–861.

Caprara A, Monaci M, Toth P, Guida PL (2006) A Lagrangian heuristic algorithm for a real-world train timetabling problem. *Discreet Appl. Math.* 154:738–753.

Chardaire P, McKeown GP, Verity-Harrison SA, Richardson SB (2005) Solving a time-space network formulation for the convoy movement problem. *Oper. Res.* 53(2):219–230.

Cordeau JF, Toth P, Vigo D (1998) A survey of optimization models for train routing and scheduling. *Transportation Sci.* 32(4):380–404.

D'Ariano A, Pacciarelli D, Pranzo M (2007) A branch and bound algorithm for scheduling trains in a railway network. *Eur. J. Oper. Res.* 183:643–657.

Dorfman MJ, Medanic J (2004) Scheduling trains on a railway network using a discrete event model of railway traffic. *Transportation Res. Part B* 38:81–98.

Dündar S, Şahin İ (2013) Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways. *Transportation Res. Part C* 27:1–15.

Higgins A, Kozan E (1998) Modeling train delays in urban networks. *Transportation Sci.* 32(4):346–357.

Higgins A, Kozan E, Ferreira L (1996) Optimal scheduling of trains on a single line track. *Transportation Res. Part B* 30:147–161.

Higgins A, Kozan E, Ferreira L (1997) Heuristic techniques for single line train scheduling. *J. Heuristics* 3:43–62.

Isaai MT, Cassaigne NP (2001) Predictive and reactive approaches to the train-scheduling problem: A knowledge management perspective. *IEEE Transportation Systems Man Cybernetics Part C* 31:476–484.

Jovanovic D, Harker PT (1991) Tactical scheduling of rail operations: The SCAN I system. *Transportation Sci.* 25(1):46–64.

Khadilkar H (2017) Data-enabled stochastic modeling for evaluating schedule robustness of railway networks. *Transportation Sci.* 51(4):1161–1176.

Krasemann JT (2012) Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances. *Transportation Res. Part C* 20:62–78.

Kroon LG, Peeters LW (2003) A variable trip time model for cyclic railway timetabling. *Transportation Sci.* 37(2):198–212.

Kwan RS, Mistry P (2003) A co-evolutionary algorithm for train timetabling. Sarker R, Reynolds R, Abbass H, Tan KC, McKay B, Essam D, Gedeon T, eds. *2003 IEEE Congress Evolutionary Comput.*, Vol. 3 (IEEE, Piscataway, NJ), 2142–2148.

Liu SQ, Kozan E (2011) Scheduling trains with priorities: A no-wait blocking parallel-machine job-shop scheduling model. *Transportation Sci.* 45(2):175–198.

Lusby RM, Larsen J, Ehrgott M, Ryan D (2011) Railway track allocation: Models and methods. *OR Spectrum* 33:843–883.

NDRC (2016) The mid-term and long-term railway network development program. National Development and Reform Commission of China, http://www.ndrc.gov.cn/zcfb/zcfbtz/201607/t20160720_811696.html.

Odijk MA (1996) A constraint generation algorithm for the construction of periodic railway timetables. *Transportation Res. Part B* 30:455–464.

Sahin I (1999) Railway traffic control and train scheduling based on inter-train conflict management. *Transportation Res. Part B* 33:511–534.

Semet Y, Schoenauer M (2005) An efficient memetic, permutation-based evolutionary algorithm for real-world train timetabling. Corne D, Michalewicz Z, McKay B, Eiben G, Fogel D, Fonseca C, Greenwood G, Raidl G, Tan KC, Zalzala A, eds. *2005 IEEE Congress on Evolutionary Comput.*, Vol. 3 (IEEE, Piscataway, NJ), 2752–2759.

Serafini P, Ukovich W (1989) A mathematical model for periodic scheduling problems. *SIAM J. Discrete Math.* 2:550–581.

Tianjin Daily (2015) The death toll of Aug. 12 massive explosion at the Port of Tianjin. (September 11), http://epaper.tianjinwe.com/qm/qm/2015-09/11/node_543.htm.

Veelenturf LP, Kidd MP, Cacchiani V, Kroon LG, Toth P (2016) A railway timetable rescheduling approach for handling large-scale disruptions. *Transportation Sci.* 50(3):841–862.

Voorhoeve M (1993) Rail scheduling with discrete sets. Technical report, Eindhoven University of Technology, Eindhoven, Netherlands.

Wagenaar JC, Kroon LG, Schmidt M (2017) Maintenance appointments in railway rolling stock rescheduling. *Transportation Sci.* 51(4):1138–1160.

Xu X, Li K, Yang L, Ye J (2014) Balanced train timetabling on a single-line railway with optimized velocity. *Appl. Math. Model.* 38:894–909.

Yang L, Zhou X, Gao Z (2014) Credibility-based rescheduling model in a double-track railway network: A fuzzy reliable optimization approach. *Omega* 48:75–93.

Zhan S, Kroon LG, Veelenturf LP, Wagenaar JC (2016) Real-time high-speed train rescheduling in case of a complete blockage. *Transportation Res. Part B* 78:182–201.

Zheng S, Kahn ME (2013) China's bullet trains facilitate market integration and mitigate the cost of megacity growth. *Proc. Natl. Acad. Sci. USA* 110:E1248–E1253.

Zheng YJ (2015) Water wave optimization: A new nature-inspired metaheuristic. *Comput. Oper. Res.* 55:1–11.

Zheng YJ, Ling HF (2013) Emergency transportation planning in disaster relief supply chain management: A cooperative fuzzy optimization approach. *Soft Comput.* 17:1301–1314.

Zheng YJ, Chen SY, Ling HF (2015a) Evolutionary optimization for disaster relief operations: A survey. *Appl. Soft Comput.* 27:553–566.

Zheng YJ, Ling HF, Xue JY (2014a) Ecogeography-based optimization: Enhancing biogeography-based optimization with ecogeographic barriers and differentiations. *Comput. Oper. Res.* 50:115–127.

Zheng YJ, Zhang MX, Zhang B (2016) Biogeographic harmony search for emergency air transportation. *Soft Comput.* 20:967–977.

Zheng YJ, Zhang MX, Ling HF, Chen SY (2015b) Emergency railway transportation planning using a hyperheuristic approach. *IEEE Trans. Intelligence Transportation Systems* 16:321–329.

Zheng YJ, Ling HF, Shi HH, Chen HS, Chen SY (2014b) Emergency railway wagon scheduling by hybrid biogeography-based optimization. *Comput. Oper. Res.* 43:1–8.

Zhou X, Zhong M (2005) Bicriteria train scheduling for high-speed passenger railroad planning applications. *Eur. J. Oper. Res.* 167:752–771.

Zhou X, Zhong M (2007) Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds. *Transportation Res. Part B* 41:320–341.