

## Research Article

# A Fast Approach for Reoptimization of Railway Train Platforming in Case of Train Delays

Yongxiang Zhang,<sup>1,2,3</sup> Qingwei Zhong,<sup>1,2,3</sup> Yong Yin ,<sup>1,2,3</sup> Xu Yan ,<sup>1,2,3</sup>  
and Qiyuan Peng<sup>1,2,3</sup>

<sup>1</sup>School of Transportation and Logistics, Southwest Jiaotong University, Chengdu 610031, China

<sup>2</sup>National United Engineering Laboratory of Integrated and Intelligent Transportation, Southwest Jiaotong University, Chengdu 610031, China

<sup>3</sup>National Engineering Laboratory of Integrated Transportation Big Data Application Technology, Southwest Jiaotong University, Chengdu 610031, China

Correspondence should be addressed to Yong Yin; yinyong@swjtu.edu.cn

Received 18 January 2020; Revised 24 February 2020; Accepted 20 May 2020; Published 3 June 2020

Academic Editor: Yu Jiang

Copyright © 2020 Yongxiang Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Train platforming is critical for ensuring the safety and efficiency of train operations within the stations, especially when unexpected train delays occur. This paper studies the problem of reoptimization of train platforming in case of train delays, where the train station is modeled using the discretization of the platform track time-space resources. To solve the reoptimization problem, we propose a mixed integer linear programming (MILP) model, which minimizes the weighted sum of total train delays and the platform track assignment costs, subject to constraints defined by operational requirements. Moreover, we design an efficient heuristic algorithm to solve the MILP model such that it can speed up the reoptimization process with good solution precision. Furthermore, a real-world case is taken as an example to show the efficiency and effectiveness of the proposed model and algorithm. The computational results show that the MILP model established in this paper can describe the reoptimization of train platforming accurately, and it can be solved quickly by the proposed heuristic algorithm. In addition, the model and algorithm developed in this paper can provide an effective computer-aided decision-making tool for the train dispatchers in case of train delays.

## 1. Introduction

Railroad transportation plays an important role in providing economic and environment-friendly transport services for passengers and goods. In particular, transport demand of railroad transportation is increasing rapidly in some parts of regional markets. For instance, according to the statistical data of the National Bureau of Statistics of China, the volume of passenger and freight traffic of China railway in 2018 has increased by 9.44% and 9.15%, compared with the volume in 2017 [1]. Thus, more and more trains are required to be operated on the railway network with limited infrastructure capacity to satisfy the large demand. Therefore, the railway operators need to adopt advanced and reliable computer-aided decision systems to improve train operation efficiency.

Moreover, the railway planning process includes the network design and line planning at the strategic level, the train timetabling, train platforming, rolling stock scheduling and crew scheduling at the tactical level, and the real-time traffic management at the operational level [2]. As a result, the core of the computer-aided decision systems is to develop effective optimization techniques for the different railway planning and operation stages. We refer the readers to Assad [3], Cordeau [4], and Lusby et al. [2] for excellent surveys on the related optimization methods. In general, train timetabling stage can determine the arrival and departure times of the trains at each of the visiting stations from an aggregate perspective [5–8]. Train operations at the stations, including the train routing on the arrival and departure routes and the train platform assignment decisions, are usually optimized

by solving the train platforming problem [8]. Apart from the boarding and alighting of passengers as well as the loading and unloading of goods at the station platforms, many other complicated operation tasks can be performed at the stations, such as the train turn-around movement and the shunting work. Thus, it is of critical importance to develop efficient and effective optimization techniques to improve the train operation efficiency at the stations without causing any conflicts.

Due to the complex layout of large railway stations and the associated dense train traffic, it is usually a challenging task for the train dispatchers to make a high-quality train operation plan within a station in a short period. Kroon et al. [9] have already proven that the train platforming problem is an NP-complete combinatorial optimization problem. As a result, many researchers have developed sophisticated and efficient models and algorithms to deal with the train platforming problem, such as the node packing model and the branch-and-cut algorithm in Zwaneveld et al. [10, 11] and the set packing model and the branch-and-price algorithm in Lusby et al. [12, 13]. In practice, however, trains may suffer from all kinds of disturbances and disruptions, such as bad weather, equipment failure, and management factors [14]. When train delays occur, the scheduled train timetable needs to be rescheduled in real time, and thus the train operations within the stations need to be reoptimized quickly to prevent any potential conflicts [15–18]. However, with regard to the train platforming problem, two managerial aspects lack in-depth study. First, most of the researchers study the train platforming problem without considering train delays, and thus they do not consider the situation where it could be impossible to make a feasible train operation plan at the stations with the given train timetable. Second, in order to generate a disposition timetable during the train rescheduling process as soon as possible, the detailed train operations at the stations are neglected, and the aggregate station capacity is adopted instead [17, 18].

In this study, we aim to bridge the research gaps lying in the train platforming and train rescheduling problems. More specifically, we reoptimize the train platforming problem in case of train delays and generate a new train operation plan within the station in real time. Our solution is to develop a mixed integer linear programming (MILP) model, where the train station is modeled using the discretized platform track time-space resources, and to propose an efficient heuristic algorithm. The goal of the proposed model and algorithm is to simultaneously minimize the deviation from the train timetable and the total train operating costs, realizing the coherence between train operations and the station management. The theoretical and practical contributions of this study mainly include the following four aspects:

- (1) The train arrival and departure times and the train platform assignment are optimized simultaneously so that the negative influence of train delays can be minimized.
- (2) The novel modeling method based on the discretized platform track time-space resources can describe the train conflicts accurately, where the complex binary

train sequencing variables in the big- $M$  modeling framework can be avoided.

- (3) An efficient heuristic algorithm is designed to quickly obtain the near-optimal solutions for the real-time reoptimization of train platforming.
- (4) A set of real-life experiments is performed to show the efficiency and effectiveness of the proposed model and algorithm.

The rest of this paper is structured as follows. Section 2 provides a comprehensive literature review on the optimization and reoptimization of train platforming problems. The method on how to describe the discretized platform track time-space resources is introduced in Section 3. In Section 4, we propose a new MILP model based on the discretized platform track time-space resources, as well as some valid equalities to strengthen the resulting MILP model. Section 5 presents the detailed algorithmic steps of a heuristic algorithm, i.e., the genetic and simulated annealing hybrid algorithm, for solving the reoptimization problem in a real-time manner. The efficiency and effectiveness of the proposed methods are verified through a set of real-life experiments in Section 6. Section 7 gives the concluding remarks and possible further research directions.

## 2. Literature Review

Train platforming problem is a key optimization stage in the railway hierarchical planning process, and it has attracted the attention of many researchers around the world [2]. Given the information of train arrival and departure times and the train running directions obtained at the train timetabling stage, the train dispatchers need to decide how to schedule the trains at the stations to avoid the train delays caused by the cross interference between train operations. The complexity of the train platforming problem grows quickly as the number of trains increases and the station layout becomes more and more complicated. Therefore, a lot of excellent works have been done regarding the train platforming problem, such as Carey [19], Zwaneveld et al. [10], Lusby et al. [12], Chakroorty and Vikram [20], Caprara et al. [21], and Kang et al. [22]. We next review these works mainly from two aspects, i.e., the train platforming problem at the tactical level and the train platforming problem considering the negative impact of train delays.

The studies on the train platforming problem at the tactical level were initially focused on scheduling the train movements at the stations efficiently and economically with the fixed or slightly relaxed train arrival and departure times. Zwaneveld et al. [10] developed the first node packing model for the train platforming problem, and the station routes were classified as the inbound route, outbound route, and complete. In addition, small deviations from the planned train arrival and departure times were considered to increase the possibility of finding a feasible solution, and an improved branch-and-cut algorithm was designed to find the optimal solutions with the goal of maximizing the number of trains routed through the station. Kroon et al. [9] proved that the node packing model in Zwaneveld et al. [10] was

NP-complete once each train had more than three routing possibilities. Based on the work in Zwaneveld et al. [10], Zwaneveld et al. [11] further proposed a weighted node packing model for the train platforming problem, where several hierarchical objectives can be considered with the unified weighted sum form. However, the node packing model has its own deficiencies to be put into practice. For instance, all of the conflicting train movements need to be identified in advance in the node packing model, and it usually results in a large-scale optimization model. As a result, Lusby et al. [12] built a set packing model of the train platforming problem, and a branch-and-price algorithm was designed such that the feasible train routes can be dynamically generated. Besides, the train platforming problem can be also reformulated as a graph colouring problem, where the number of colours is determined by the available tracks in the stations [23]. An integer programming model that was equivalent to graph colouring problem in Cardillo and Mione [23] was developed by Billionnet [24]. Caprara et al. [21] also allowed a slight relaxation on the ideal arrival and departure times of the trains at the platforms, and they presented a 0-1 quadratic programming model to solve the train platforming problem. In particular, the quadratic term in the objective function was designed to consider the soft incompatibility costs between train paths. Sels et al. [25] developed a new MILP model to schedule the trains in the current and future train set, and penalty costs were considered in the objective when a train was cancelled or assigned to a different platform.

Apart from the exact solution methods, some researchers also developed heuristic algorithms for the train platforming problem [26–29]. Carey and Carville [26] introduced a sequential construction heuristic algorithm for the train platforming problem at a single station, which was designed according to the train dispatchers' manual methods. In addition, trains can be delayed in the planning process to generate a feasible solution, and the assignment of trains to the undesired platforms was penalized. Carey and Crawford [27] further extended the work in Carey and Carville [26] by considering the train scheduling and platforming problems at multiple stations, where those stations were connected by several one-way lines. Similarly, trains can be also delayed when they were scheduled from one station to the corresponding next station. Kang et al. [28] proposed a simulated annealing heuristic algorithm with the goal of achieving the balanced occupation of station turnouts while minimizing the total occupation times of the station turnouts. Wu et al. [29] developed a mean-variance optimization model for the train platforming problem based on Markowitz's portfolio theory, which was aimed to minimize the total occupation time costs of groups of turnouts. Moreover, an efficient simulated annealing heuristic algorithm was designed to solve the resulting mixed integer nonlinear programming (MINLP) model.

Some researchers also tried to integrate the two stages of train timetabling and train platforming problems to further increase the railway network capacity utilization. Carey [19] proposed an integrated optimization model to simultaneously optimize the lines, platforms, and routes of the trains in a railway network, and an improved sequential

searching method was adopted to solve the resulting model. In addition, the arrival and departure headways between two trains were enforced to guarantee the train running safety. Dorfman and Medanic [30] proposed a discrete event model to schedule the trains in the railway network, where the train routes were assumed to be fixed for simplicity. D'ariano et al. [31] formulated the train scheduling problem through the alternative graph modeling approach, and an efficient branch-and-bound algorithm was designed to solve the MILP model. Rodriguez [32] proposed a combined constraint programming and simulation method for scheduling trains in the junctions, where the constraint programming model representing the signalling system behaviour and the simulation model can reflect the train and driver behaviour. More recently, D'Ariano et al. [33] and Zhang et al. [8] studied the integrated train timetabling and maintenance task scheduling problems at the microscopic level, and efficient heuristic algorithms based on the commercial solvers were developed accordingly.

Some works were also performed on the optimization of a delay resistant train platforming plan and the real-time reoptimization of train platforming problem in case of unexpected train delays. Herrman [34] studied the stable train routing problem, where the goal was to keep the train schedule to be conflict-free as much as possible in case of perturbations. In addition, a local search heuristic approach was designed to solve the train schedule stabilization problem. Bešinović and Goverde [35] proposed a max-plus automata model for the station capacity assessment and a delay propagation model for the robustness evaluation of the train platforming plan. Based on these two models, the objectives of minimization of capacity occupation and maximization of robustness were then achieved with a heuristic approach. Kang et al. [22] introduced the concept of stochastic schedule with uncertain train arrival and departure times, and a simulated annealing algorithm was developed to solve the corresponding train platforming problem. D'Ariano et al. [36] designed a rolling horizon approach for the real-time train rescheduling problem, and the alternative graph formulation was adopted for solving the smaller subproblem in each rolling period. Chakraborty and Vikram [20] developed a MILP model for optimally allocating trains to the platform tracks, where the accurate train arrival times can only be available shortly before the train arrives at the station such that trains could be reassigned to different platforms. Besides, the headway between two trains was also considered while delaying the train arrival and departure times. Lusby et al. [13] proposed a set packing model for rescheduling the trains in the junctions in case of disruptions. In particular, a branch-and-price algorithm was introduced to solve the set packing model, and the objective was to minimize the weighted deviations from the trains' scheduled arrival times. Liu et al. [37] developed a MINLP model to reallocate the tracks of trains at the stations in a real-time manner, where the goal was to minimize average use time of groups of turnouts without changing the planned train times. Moreover, a genetic simulated annealing algorithm was also designed to solve the MINLP model efficiently.

Table 1 summarizes the modeling approaches, objectives, solution methods, and other characteristics of relevant publications on the optimization and reoptimization of train platforming problems, and major highlights can be drawn from Table 1. First, most of the studies focus on the train platforming problem at the tactical level, and the scheduled train arrival and departure times at the stations are usually assumed to be fixed. Therefore, the optimization objectives of those studies are generally minimizing the total costs or maximizing the total benefits. Second, exact algorithms account for the majority of the solution methods to solve the optimization or reoptimization of the train platforming problems. Therefore, efficient and flexible heuristic algorithms are required, especially for the real-time reoptimization of the train platforming problem. Third, among the studies that consider the negative influence of train delays, only one research by Chakroborty and Vikram [20] simultaneously incorporates the factors of train delay, reassignment of platform, and deviations from scheduled train arrival and departure times. However, they adopt a commercial solver to deal with the MILP model. In this study, we develop a new MILP model for the reoptimization of the train platforming problem based on the discretized time-space resources, and we also design an efficient heuristic algorithm to guarantee the achievement of real-time reoptimization.

### 3. Analysis of Platform Track Time-Space Resources

Let  $T$  be the planning horizon; we handle the time resources as small time units of length  $\Delta\tau$ . The number of time units is equal to  $|T| = \lceil T/\Delta\tau \rceil$  in the entire planning horizon. In addition, the number of platform tracks in a station is denoted by  $|I|$ , i.e., the maximum spatial capacity. Hence, the platform track time-space resources of a station can be represented by a two-dimensional matrix  $R$ :

$$R = [r_{i,t}] = \begin{bmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,|T|} \\ r_{2,1} & r_{2,2} & \dots & r_{2,|T|} \\ \dots & \dots & \dots & \dots \\ r_{|I|,1} & r_{|I|,2} & \dots & r_{|I|,|T|} \end{bmatrix}, \quad (1)$$

where  $i$  and  $t$  are the indexes of the platform track and the time unit, respectively, and a binary variable  $r_{i,t}$  in matrix  $R$  denotes the occupation and vacancy state of the platform track time-space resource  $(i, t)$ , where

$$r_{i,t} = \begin{cases} 1, & \text{platform track time - space resource } (i, t) \text{ is used,} \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Figure 1 provides an illustrative example of the modeling method of platform track time-space resources. Suppose that 5 trains successively arrive at or depart from station  $N$  which has 6 platform tracks within the planning horizon of 60 min. The detailed schedules of train operations in both directions are given in Figures 1(b) and 1(c), and the time unit  $\Delta\tau$  is set to 5 min. Platform track time-space resources and the

corresponding matrix  $R$  of a feasible usage plan are described in Figures 2(a) and 2(b), respectively. The application requirements of the time-space resources modeling method for the reoptimization of train platforming problem can be formulated as follows:

- (1) *Inseparability*. A train must occupy only one platform track and cannot occupy more than one platform track simultaneously.
- (2) *Exclusivity*. One platform track can only store one train at any time unit to prevent the train conflicts.
- (3) *Continuity*. A train operation on one platform track with the duration equal to  $\Delta t$  time units cannot be interrupted. If one train starts to use track  $i$  at time  $t$ , it continues to occupy the platform track  $i$  until  $t + \Delta t$ , i.e.,  $r_{i,t} = r_{i,t+1} = r_{i,t+2} = \dots = r_{i,t+\Delta t} = 1$ .

## 4. Mathematical Modeling for Reoptimization of Train Platforming

*4.1. Modeling Assumptions.* Without loss of generality, we make the following key modeling assumptions to facilitate the modeling process.

- (1) When it is impossible to eliminate the conflicts through reassignment of platforms, the train arrival and departure times can only be delayed to resolve the conflicts.
- (2) One minute is applied to discretize the platform track time-space resources, and smaller time units can be adopted if necessary.
- (3) Trains are modeled as single objects moving in the railway station, and thus the influence of train lengths on the clearing times of the platform tracks is not considered.
- (4) By enforcing the arrival and departure headways between two trains running in the same direction as well the safety time interval for platform track operation, it can be guaranteed that there are no conflicts between any two trains.
- (5) All trains will pass through the railway station, and turn-around movements of the trains are not considered for simplicity.

*4.2. Definitions of Sets, Parameters, and Variables.* Sets, parameters, and variables of this study are defined in Table 2. In particular, we assume that all parameter values related to time are multiples of the time unit  $\Delta\tau$ .

*4.3. Objective Function.* The objective function in equation (3) contains the weighted sum of two parts of costs. The first part is the sum of train arrival and departure delays, considering the train priority  $P_l$  and the weighting factor  $\alpha$ . The second part is the total platform track assignment costs. In particular, the more that train  $l$  is preferred for platform track  $i$ , the smaller is the corresponding platform track assignment cost  $c_{l,i}$ . Note that value of  $c_{l,i}$  can be determined

TABLE 1: Summary of relevant publications on the optimization and reoptimization of the train platforming problems.

Publication	Modeling approach	Objective	Solution method	Train delay	Reassignment of platform	Deviations from scheduled train times	Largest instance solved	Computation times (in seconds) of the largest instance
Zwaneveld et al. [10]	BILP	Max the number of routed trains	Branch-and-cut			√	A railway station with 17 platforms and 27 trains	85
Cardillo and Mione [23]	GC	Feasibility	Iterative heuristic algorithm				A railway station with 16 tracks and 242 trains	0.67
Billionnet [24]	BILP	Feasibility	Optimization software				A railway station with 14 tracks and 200 trains	1.64
Carey and Carville [26]	MILP	Min the costs of deviations from preferred times, platforms, and lines	Iterative sequential searching method		√	√	A railway station with 29 subproblems and 491 trains	—
Rodriguez [32]	CP	Min train delays	Tree search	√		√	A 15.18 km long railway junction with 24 trains	54.76
Lusby et al. [12]	BILP	Max total benefits	Branch-and-price				A railway junction with 64 block sections and 45 trains	26.4
Caprara et al. [21]	BQP	Min total costs	Branch-and-cut-and-price				A railway station with 14 platforms and 237 trains	350
Sels et al. [25]	MILP	Min total penalties	Commercial solvers		√		A railway station with 14 platforms and 160 trains	3.0
Zhang et al. [8]	MILP	Min total train journey times and maintenance tardiness costs	Iterative heuristic algorithm				A railway network with 27 stations, 55 segments, and 58 trains	7200
Liu et al. [37]	MINLP	Min average use time of groups of turnouts	Genetic simulated annealing algorithm		√		A railway station with 11 tracks and 30 trains	1800
Kang et al. [22]	MILP	Min total occupation times and balanced track occupation	Simulated annealing	√			A railway station with 11 tracks and 28 trains	726
Chakroborty and Vikram [20]	MILP	Min total costs	Commercial solver	√	√	√	A railway station with 9 platforms and 110 trains	600
Lusby et al. [13]	BILP	Min total deviations	Branch-and-price	√		√	A railway junction with 524 track sections and 66 trains	151.55

TABLE 1: Continued.

Publication	Modeling approach	Objective	Solution method	Train delay	Reassignment of platform	Deviations from scheduled train times	Largest instance solved	Computation times (in seconds) of the largest instance
This paper	MILP	Min total costs	Genetic and simulated annealing hybrid algorithm	√	√	√	A railway station with 11 platforms and 70 trains	32.92

Note. BILP represents 0-1 integer linear programming; GC represents graph colouring; MILP represents mixed integer linear programming; CP represents constraint programming; BQP represents 0-1 quadratic programming; “—” represents no specific information is available.

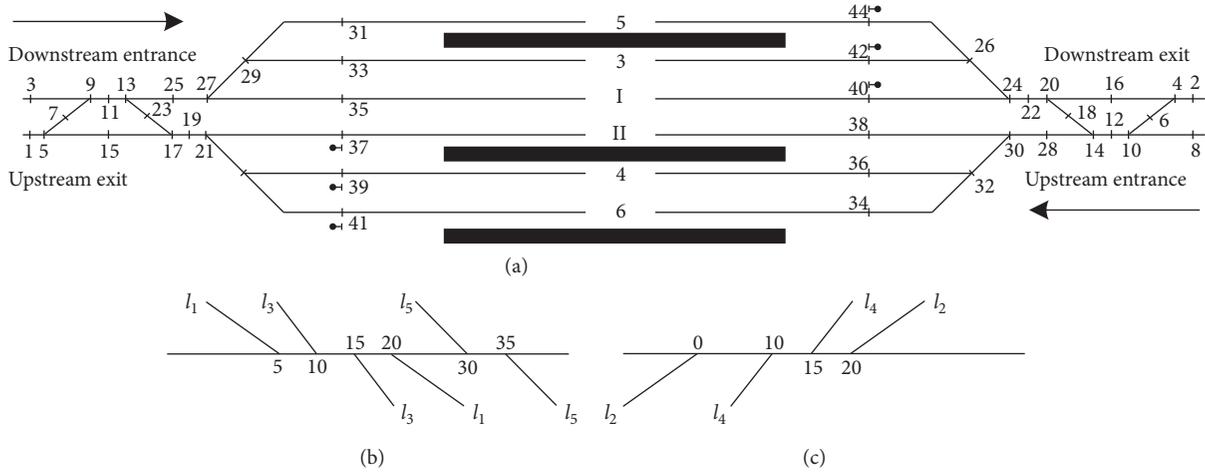


FIGURE 1: Layout and train timetables of station N: (a) layout of station N; (b) train diagram of station N in the downstream direction; (c) train diagram of station N in the upstream direction.

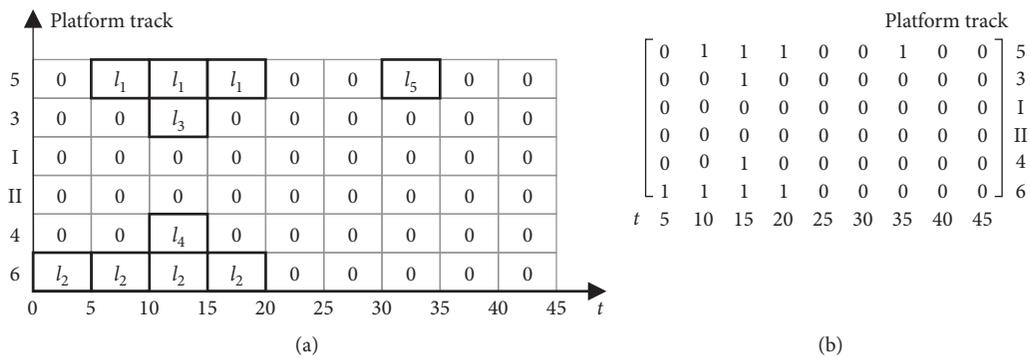


FIGURE 2: Platform track time-space resource usage plan and time-space resource matrix: (a) usage of platform track time-space resources; (b) platform track time-space resource matrix.

according to several influence factors, including the initial platform track assignment plan, the convenience of passenger boarding and alighting, and other special requirements for train operations. Therefore, the smaller the value of  $c_{l,i}$  is, the more the train  $l$  prefers the platform track  $i$ .

$$\min z = \alpha \sum_{l \in L} P_l [(y_{l,a} - t_{l,a}) + (y_{l,d} - t_{l,d} - D)] + \sum_{l \in L} \sum_{i \in I_l} w_{l,i} c_{l,i}. \tag{3}$$

#### 4.4. Constraints

**4.4.1. Relation between Platform Track State Variables.** According to definitions of platform track state variables  $x_{l,i,t}$ ,  $u_{l,i,t}$ , and  $v_{l,i,t}$ , constraint (4) expresses the relationship among those three platform track state variables. In particular, when both the variables  $u_{l,i,t}$  and  $v_{l,i,t}$  are equal to 0, it can be known that train  $l$  is occupying the platform track  $i$  at time  $t$ , i.e.,  $x_{l,i,t} = 1$ . Figure 3 shows an illustrative example on how to calculate the value of  $x_{l,i,t}$  according to the values of the variables  $u_{l,i,t}$  and  $v_{l,i,t}$ . More specifically, train  $l_1$  arrives at the platform track 5 in Figure 1 at time 5 and it leaves from the corresponding platform track at time 20. Therefore, Figures 3(a) and 3(b) present the values of the variables  $u_{l,i,t}$  and  $v_{l,i,t}$  for train  $l_1$  according to its actual arrival and departure times. In addition, according to constraint (4), the values of  $v_{l,i,t}$  in Figure 3(c) can be easily obtained.

$$x_{l,i,t} = 1 - (u_{l,i,t} + v_{l,i,t}), \quad \forall l \in L, i \in I, t \in \{1, 2, \dots, MT\}. \quad (4)$$

**4.4.2. Actual Train Arrival and Departure Times.** Constraints (5) and (6) show that values of actual arrival time  $y_{l,a}$  and actual departure time  $y_{l,d}$  of train  $l$  can be inferred from the values of platform track utilization state variable  $u_{l,i,t}$  and the platform track clearance state variable  $v_{l,i,t}$ .

$$y_{l,a} = MT - \sum_{i \in I} \sum_{t=1}^{MT} (1 - u_{l,i,t}), \quad \forall l \in L, \quad (5)$$

$$y_{l,d} = MT - \sum_{i \in I} \sum_{t=1}^{MT} v_{l,i,t}, \quad \forall l \in L. \quad (6)$$

**4.4.3. Platform Track Assignment Constraint.** Constraint (7) requires that each train  $l$  can only be assigned to one platform track, and the corresponding platform track has to be in the set  $I_l$ . Furthermore, constraint (8) enforces that a train  $l$  cannot be assigned to forbidden platform tracks, i.e., tracks that are not in the set  $I_l$ . Constraints (7)-(8) provide a flexible method to restrict the platform track choices of the trains.

$$\sum_{i \in I_l} w_{l,i} = 1, \quad \forall l \in L, \quad (7)$$

$$w_{l,i} = 0, \quad \forall l \in L, i \in I \setminus I_l. \quad (8)$$

**4.4.4. Platform Track Capacity Constraint.** Constraint (9) ensures that any platform track  $i$  can only be occupied by at most one train at any time  $t$ , such that no conflicts can occur between any two trains that are assigned to the same platform track.

$$\sum_{l \in L} x_{l,i,t} \leq 1, \quad \forall i \in I, t \in \{1, 2, \dots, MT\}. \quad (9)$$

**4.4.5. Continuous Train Operation on the Platform Track.** Constraints (10)–(12) guarantee that train operations on the platform tracks should be performed continuously, by enforcing the relationship between the values of variables  $u_{l,i,t}$  and  $v_{l,i,t}$ . For instance, if  $w_{l,i} = 1$  in constraint (10), which implies that train  $l$  is assigned to the platform track  $i$ , constraint (10) changes to  $u_{l,i,t} \geq u_{l,i,t+1}$ . According to the definition of  $u_{l,i,t}$  in Section 4.2, the condition  $u_{l,i,t} \geq u_{l,i,t+1}$  is obviously satisfied. On the other hand, if  $w_{l,i} = 0$  in constraint (10), constraint (10) is no longer valid. Similarly, constraints (11) and (12) define the relationships between  $v_{l,i,t}$  and  $v_{l,i,t+1}$  as well as between  $u_{l,i,t}$  and  $u_{l,i,t+1}$ .

$$u_{l,i,t} \geq u_{l,i,t+1} + w_{l,i} - 1, \quad \forall l \in L, i \in I, t \in \{1, 2, \dots, MT - 1\}, \quad (10)$$

$$v_{l,i,t} \leq v_{l,i,t+1} - w_{l,i} + 1, \quad \forall l \in L, i \in I, t \in \{1, 2, \dots, MT - 1\}, \quad (11)$$

$$u_{l,i,t} \leq u_{l,i,t+1} + w_{l,i}, \quad \forall l \in L, i \in I, t \in \{1, 2, \dots, MT - 1\}. \quad (12)$$

**4.4.6. Safety Headway between Two Trains.** Constraints (13)–(18) impose the required safety headway between two arrival or departure trains running in the same direction, and the train sequencing variables  $\lambda_{l,k}$  and  $\mu_{l,k}$  as well as the platform track choice variables  $w_{l,i}$  and  $e_{l,k}$  are also embedded in those constraints. Constraint (13) enforces the safety headway between two arrival trains in the same direction, and it is valid only if  $\lambda_{l,k} = 1$ , i.e., train  $l$  arrives at the station before train  $k$ . When  $\lambda_{l,k} = 1$  in constraint (13), it can be shown that the safety headway between trains  $l$  and  $k$  depends on the value of the variable  $e_{l,k}$ . More specifically, if  $e_{l,k} = 1$ , trains  $l$  and  $k$  are assigned to the same platform track, and they are separated by the safety time interval  $D$ . Otherwise, trains  $l$  and  $k$  are assigned to two different platform tracks, and they are separated by the arrival headway  $h_a$ . Meanwhile, constraint (14) guarantees the safety headway between two departure trains in the same direction. Constraint (15) ensures that if both the variables  $w_{l,i}$  and  $w_{k,i}$  are equal to 1, the value of the variable  $e_{l,k}$  can be enforced to be 1. Furthermore, constraint (16) describes that the values of the two variables  $e_{l,k}$  and  $e_{k,l}$  for two different trains  $l$  and  $k$  are equal. Besides, constraints (17) and (18) guarantee that only one of the train sequencing variables  $\lambda_{l,k}$  and  $\lambda_{k,l}$  or  $\mu_{l,k}$  and  $\mu_{k,l}$  between any two different trains is equal to 1.

$$y_{l,a} - y_{k,a} \geq (1 - e_{l,k})h_a + e_{l,k}D - \lambda_{l,k}M, \quad (13)$$

$$\forall l, k \in L: l \neq k, \pi_l = \pi_k,$$

$$y_{l,d} - y_{k,d} \geq (1 - e_{l,k})h_d + e_{l,k}D - \mu_{l,k}M, \quad (14)$$

$$\forall l, k \in L: l \neq k, \pi_l = \pi_k,$$

TABLE 2: Sets, parameters, and variables.

Symbol	Definition
<b>Sets</b>	
$L$	Set of trains, indexed by $l$
$L_1$	Set of delayed trains, $L_1 \subseteq L$
$I$	Set of platform tracks, indexed by $i$
$I_l$	Set of platform tracks that train $l$ can utilize, $I_l \subseteq I$
$T$	Set of time intervals in the planning horizon, indexed by $t$
<b>Parameters</b>	
$c_{l,i}$	Cost of assigning train $l$ to platform track $i$
$\pi_l$	0-1 parameter is equal to 1 if train $l$ is running in the downstream direction; 0, otherwise
$q_{l,i}$	0-1 parameter is equal to 1 if train $l$ was initially assigned to platform track $i$ before a delay occurs; 0, otherwise
$S$	The time when the information of train delays is updated
$t_{l,a}$	Scheduled arrival time of train $l$
$t_{l,d}$	Scheduled departure time of train $l$
$t_{l,a}^1$	Estimated arrival time of train $l$ when a delay occurs
$t_{l,d}^1$	Estimated departure time of train $l$ when a delay occurs
$\Delta_l$	Dwell time of train $l$
$P_l$	Priority of train $l$
$D$	Safety time interval for platform track operation
$MT$	Sum of the length of the planning horizon $T$ and the safety time interval for platform track operation $D$
$\Delta_{\max}$	Maximum value of dwell times of all trains
$h_a$	Headway between two arrival trains running in the same direction
$h_d$	Headway between two departure trains running in the same direction
$\alpha$	Objective function weighting factor
$M$	A sufficiently large number
<b>Variables</b>	
$w_{l,i}$	0-1 platform track assignment variable = 1 if train $l$ is assigned to platform track $i$ ; = 0 otherwise
$e_{l,k}$	0-1 same platform track choice variable = 1 if train $l$ and train $k$ choose the same platform track; = 0 otherwise
$x_{l,i,t}$	0-1 platform track occupancy state variable = 1 if train $l$ occupies platform track $i$ at moment $t$ ; = 0 otherwise
$u_{l,i,t}$	0-1 platform track utilization state variable = 1 if train $l$ has not yet arrived at platform track $i$ at moment $t$ ; = 0 otherwise
$v_{l,i,t}$	0-1 platform track clearance state variable = 1 if train $l$ has left platform track $i$ at moment $t$ ; = 0 otherwise
$\lambda_{l,k}$	0-1 train arrival sequence variable = 1 if train $l$ arrives at the station before train $k$ ; = 0 otherwise
$\mu_{l,k}$	0-1 train departure sequence variable = 1 if train $l$ departs from the station before train $k$ ; = 0 otherwise
$y_{l,a}$	Actual arrival time of train $l$ at the station
$y_{l,d}$	Actual departure time of train $l$ at the station

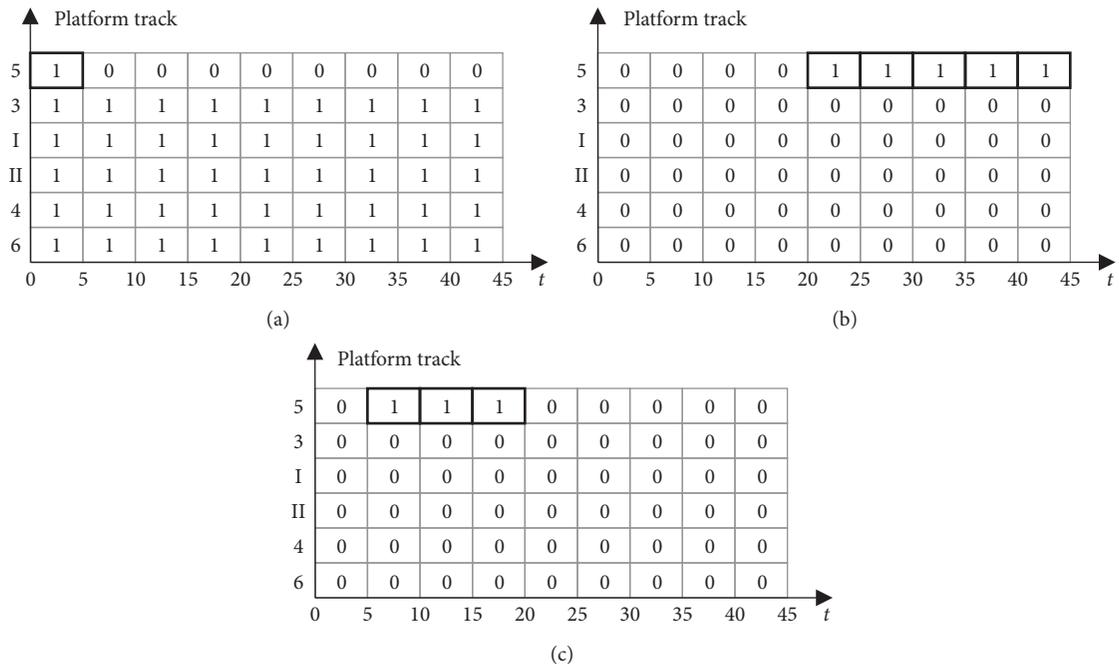


FIGURE 3: An illustrative example showing the values of platform track state variables  $u_{l,i,t}$ ,  $v_{l,i,t}$ , and  $x_{l,i,t}$ : (a) values of the variable  $u_{l,i,t}$ ; (b) values of the variable  $v_{l,i,t}$ ; (c) values of the variable  $x_{l,i,t}$  and  $x_{l,i,t} = 1 - (u_{l,i,t} + v_{l,i,t})$ .

$$e_{l,k} \geq w_{l,i} + w_{k,i} - 1, \quad \forall l, k \in L, i \in I_l \cap I_k: k > l, \pi_l = \pi_k, \quad (15)$$

$$e_{l,k} = e_{k,l}, \quad \forall l, k \in L: k > l, \pi_l = \pi_k, \quad (16)$$

$$\lambda_{l,k} + \lambda_{k,l} = 1, \quad \forall l, k \in L: k > l, \pi_l = \pi_k, \quad (17)$$

$$\mu_{l,k} + \mu_{k,l} = 1, \quad \forall l, k \in L: k > l, \pi_l = \pi_k. \quad (18)$$

**4.4.7. Minimum Train Dwell Time Constraint.** Constraint (19) enforces the minimum dwell time for each train  $l$ . In particular, the right-hand side of constraint (19) can only be valid when  $w_{l,i} = 1$ , i.e., train  $l$  is assigned to the platform track  $i$ . Note that the safety time interval for platform track operation  $D$  is also included in the right-hand side of the constraints so that the required safety time interval for trains assigned to the same platform track is imposed. In addition, the minimum dwell time  $\Delta_l$  of a train  $l$  is a deterministic value.

$$\sum_{t=1}^{MT} x_{l,i,t} \geq w_{l,i} (\Delta_l + D), \quad \forall l \in L, i \in I. \quad (19)$$

**4.4.8. Actual Train Arrival Time and Departure Time Constraint.** Constraints (20)–(22) specify that the actual arrival and departure times of the trains should be no less than the corresponding planned arrival and departure times, respectively. Note that safety time interval  $D$  is introduced into the right-hand sides of constraints (21)–(22), such that the safety time interval  $D$  for platform track operation is implicitly satisfied between the departure time  $y_{l,d}$  of train  $l$  and the arrival time of any other trains that are assigned to the same platform track.

$$y_{l,a} \geq t_{l,a}, \quad \forall l \in L, \quad (20)$$

$$y_{l,d} \geq t_{l,d} + D, \quad \forall l \in L, \quad (21)$$

$$y_{l,d} \geq y_{l,a} + \Delta_l + D, \quad \forall l \in L. \quad (22)$$

**4.4.9. Domain of Variables.** Constraints (23)–(25) define the domain of variables. Note that the variables  $x_{l,i,t}$ ,  $y_{l,a}$ , and  $y_{l,d}$  are intermediate variables to facilitate the model building process, and their values can be directly inferred from the variables  $u_{l,i,t}$  and  $v_{l,i,t}$ .

$$w_{l,i} = \{0, 1\}, \quad \forall l \in L, i \in I, \quad (23)$$

$$u_{l,i,t}, v_{l,i,t} = \{0, 1\}, \quad \forall l \in L, i \in I, t \in \{1, 2, \dots, MT\}, \quad (24)$$

$$e_{l,k}, \lambda_{l,k}, \mu_{l,k} = \{0, 1\}, \quad \forall l, k \in L: l \neq k, \pi_l = \pi_k. \quad (25)$$

In short, formulations (3)–(25) constitute the complete MILP model for the reoptimization of train platforming problem. Due to the complicated filter conditions for the

ranges of constrains (13)–(18), we simply approximate the number of constraints (13)–(18) without considering those filter conditions. As a result, the upper bound on the number of constraints of the MILP model is equal to  $MT|I|(2|L| + 1) + 6|L| + |L|^2(5 + |I|) + \sum_{l \in L} (|I| - |I_l|)$ , which increases polynomial as the number of trains  $|L|$ , the number of platform tracks  $|I|$ , and the length of planning horizon  $T$  increase. Similarly, the upper bound on the number of variables is equal to  $|L| \cdot |I|(1 + MT) + |L|^2$ .

**4.5. Valid Inequalities.** Valid equalities are additional constraints that have been implicitly satisfied according to other necessary model constraints (4)–(25). However, they can be applied to strengthen the model formulation [17, 38]. Constraints (26)–(29) define the four valid inequalities that are related to the variables  $u_{l,i,t}$ ,  $v_{l,i,t}$ ,  $x_{l,i,t}$ , and  $w_{l,i}$ . Note that the valid inequalities are only implemented in a commercial solver to speed up the solving process of the commercial solver.

$$u_{l,i,t} \geq 1 - w_{l,i}, \quad \forall l \in L, i \in I, t \in \{1, 2, \dots, MT\}, \quad (26)$$

$$v_{l,i,t} \leq w_{l,i}, \quad \forall l \in L, i \in I, t \in \{1, 2, \dots, MT\}, \quad (27)$$

$$x_{l,i,t} \leq w_{l,i}, \quad \forall l \in L, i \in I, t \in \{1, 2, \dots, MT\}, \quad (28)$$

$$x_{l,i,t} = 0, \quad \forall l \in L, i \in I, t \in \{1, 2, \dots, t_{l,a} - 1\} \vee t \in \{t_{l,d} + \Delta_{\max} + D + 1, \dots, MT\}. \quad (29)$$

The principles of building valid inequalities (26)–(29) are similar. For instance, in valid inequality (26), if train  $l$  occupies platform track  $i$ , then valid inequality (26) is equivalent to  $u_{l,i,t} \geq 0$  which turns out to be ineffective. However, if train  $l$  does not occupy the platform track  $i$ , then valid inequality (26) is equivalent to  $u_{l,i,t} \geq 1$  which implies  $u_{l,i,t} = 1$ . Valid inequality (29) considers when the station capacity is not sufficient and two conflicting trains need to be assigned to the same platform track, then one of the two trains with lower priority can be delayed at most by  $\Delta_{\max}$ , which means  $x_{l,i,t}$  can be constrained to 0 when  $t \in \{1, 2, \dots, t_{l,a} - 1\}$  or  $t \in \{t_{l,d} + \Delta_{\max} + D + 1, \dots, MT\}$ . Overall, a total of  $4MT|L| \cdot |I| - \sum_{l \in L} |I|(t_{l,d} + \Delta_{\max} + D - t_{l,a} + 1)$  additional constraints are added into the model after introducing the valid inequalities (26)–(29).

## 5. Genetic and Simulated Annealing Hybrid Algorithm

In order to recover the train operations as soon as possible in case of train delays, a genetic and simulated annealing hybrid algorithm (GSAHA) is designed to solve the reoptimization model efficiently and effectively [37, 39, 40]. In particular, the simulated annealing algorithm (SA) has good ability in jumping out of local optimal solutions, and its convergence performance is robust against the generated initial solutions. However, the convergence speed of SA is generally slow. On

the other hand, the genetic algorithm (GA) is a parallel search algorithm that usually converges very fast, while SA is more likely to be trapped into local optimal solutions and it is not robust against the generated initial solutions. As a result, the GSAHA algorithm combines the advantages of GA and SA. Therefore, GSAHA is robust on the convergence performance while avoiding being trapped into the local optimal solutions [37, 40, 41]. Figure 4 demonstrates the solution framework of the GSAHA algorithm, including the section names corresponding to some key algorithmic steps. The implementation details for the components of GSAHA are illustrated as follows.

**5.1. Chromosome Representation.** Figure 5 shows the one-dimensional real-value encoding method that is used to represent chromosomes. Each chromosome denotes a platform track assignment plan. In particular, if the value of the  $l^{\text{th}}$  gene is equal to  $i$ , the  $l^{\text{th}}$  train is assigned to platform track  $i$  with its scheduled arrival and departure times. Furthermore, the length of each chromosome is equal to the number of trains  $|L|$ , and the genes of a chromosome are numbered in descending order according to the scheduled arrival times of trains. Moreover, the value range of each gene is located within the range  $[1, |I|]$ , and thus there could be  $|I|^{|L|}$  possible chromosomes in total.

**5.2. Generating Initial Population.** Considering diversity and rationality of individuals in the initial population, the following strategies are proposed to generate the initial population.

Step 1. Denote platform tracks whose numbers are smaller than the number of platform tracks  $|I|$  as the set  $I_1$ .

Step 2. Select  $\lfloor |L| / (|I| - 1) \rfloor$  trains that have not been selected yet and assign those trains to one of the unassigned platform tracks in set  $I_1$ .

Step 3. Repeat Step 1 until all platform tracks in set  $I_1$  are assigned and assign the remaining  $|L| - \lfloor |L| / (|I| - 1) \rfloor \cdot (|I| - 1)$  trains to the last platform track numbered as  $|I|$ .

Step 4. Repeat Step 2 and Step 3 until all individuals in the initial population are generated.

**5.3. Obtaining Feasible Solutions.** The chromosome designed in Section 5.1 only assigns trains to the platform tracks, i.e., to determine the platform track spatial resources that each train occupies. However, it is still possible that two trains may conflict with each other on the occupation of platform track temporal resources due to the violation of safety headway requirements, including the safety time interval between two trains assigned to the same platform track  $D$ , headway between two arrival trains running in the same direction  $h_a$ , and headway between two departure trains running in the same direction  $h_d$ . Hence, a heuristic rule is designed to resolve the temporal conflicts according to the constraints defined in Section 4.5.

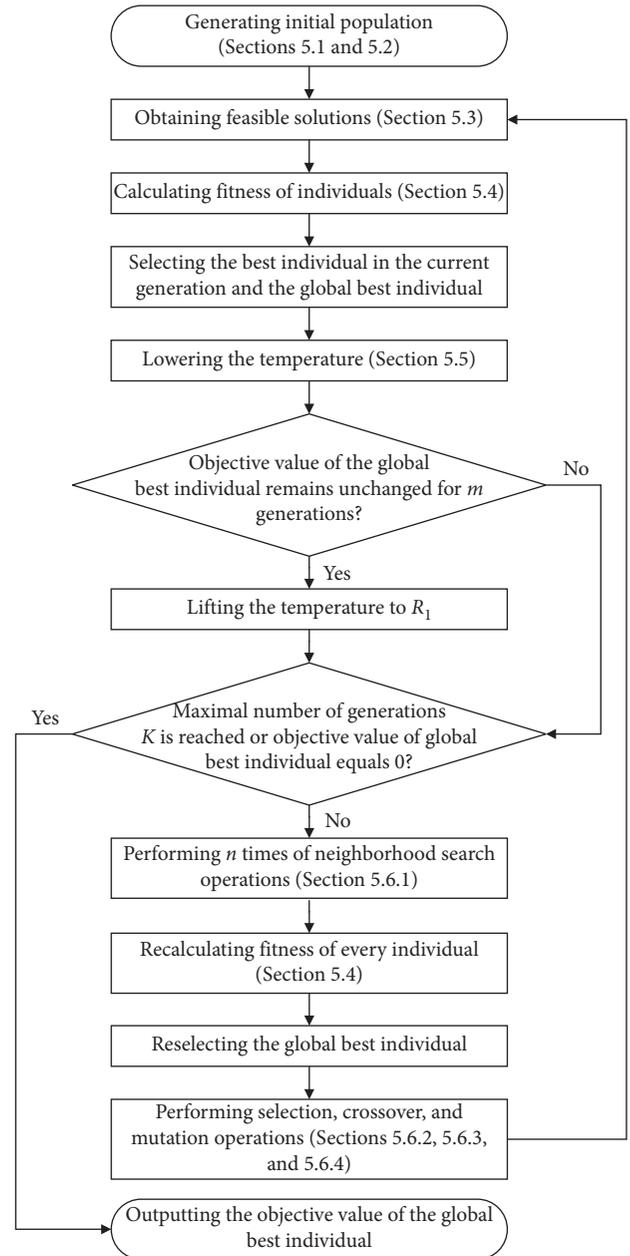


FIGURE 4: Solution framework of the GSAHA algorithm.

Platform track	5	2	3	6	...	4	7	1
Train	1	2	3	4	...	$ L -2$	$ L -1$	$ L $

FIGURE 5: Illustration of the chromosome representation.

**5.4. Fitness Function.** The fitness function in equation (30) is designed to evaluate each individual such that the algorithm can achieve a better convergence performance.

$$f_i(r_k) = \exp \left\{ -\frac{f(i) - f_{\min}}{r_k} \right\}, \quad (30)$$

where  $r_k$  represents the temperature at the  $k^{\text{th}}$  generation,  $f(i)$  represents the objective value of the  $i^{\text{th}}$  chromosome,

**For** each train  $i$  ( $1 \leq i \leq |L|$ )  
**For** each train  $j$  ( $1 \leq j < i$ )  
**If** train  $i$  conflicts with train  $j$   
    Fix the arrival and departure times of train  $i$  and record the weighted sum delay amount  $\vartheta_i$  after resolving the conflicts of trains numbered before train  $i$ ;  
    Fix the arrival and departure times of train  $j$  and record the weighted sum delay amount  $\vartheta_j$  after resolving the conflicts of trains numbered before train  $i$ ;  
    **If**  $\vartheta_i \leq \vartheta_j$   
        Adopt the adjust method by fixing the arrival and departure times of train  $i$ ;  
    **Else**  
        Adopt the adjust method by fixing the arrival and departure times of train  $j$ ;  
    **End If**  $\vartheta_i \leq \vartheta_j$   
**End If** train  $i$  conflicts with train  $j$   
**End For** each train  $j$  ( $1 \leq j < i$ )  
**End For** each train  $i$  ( $1 \leq i \leq |L|$ )

Step 5. Sort all trains in descending order by their scheduled or estimated arrival times and number them from 1 to  $|L|$ .  
Step 6. Use Algorithm 1 to resolve the temporal conflicts between any two trains in the order given in Step 1. Note that Algorithm 1 will not lead to a deadlock between trains where trains can always be delayed to resolve the temporal conflicts.  
Step 7. Calculate the weighted sum of arrival and departure delays compared to the corresponding scheduled or estimated train arrival and departure times. This operation considers all trains in set  $L$  and the platform track assignment costs, and thus the value calculated during this step serves as the objective value of the chromosome.

ALGORITHM 1: A heuristic method to resolve the temporal conflicts between trains with given train order.

$f_{\min}$  represents the minimal objective value at the  $k^{\text{th}}$  generation, and  $f_i(r_k)$  represents fitness value of the  $i^{\text{th}}$  chromosome when the temperature is  $r_k$ . Fitness function in equation (30) is an important feature of the SA algorithm, and it has a good capacity to accelerate the convergence of the algorithm.

**5.5. Temperature Decline Function.** After determining the initial temperature  $R$ , the temperature decline function in equation (31) is used to lower the temperature at each iteration.

$$r_k = R \cdot \eta^k, \quad (31)$$

where  $r_k$  represents the temperature at the  $k^{\text{th}}$  generation and the constant  $\eta$  represents the temperature decline rate.

## 5.6. Genetic Operators

**5.6.1. Neighborhood Search.** Neighborhood search operator is applied to every chromosome. For instance, neighborhood search operator modifies the value of one gene in chromosome  $i$  randomly to generate a new chromosome  $j$ , and the objective value  $f(j)$  of chromosome  $j$  is recalculated. Chromosome  $j$  is accepted or rejected to replace chromosome  $i$  according to the probability  $P_{ij}(r_k)$  in the following equation:

$$P_{ij}(r_k) = \min \left\{ 1, \exp \left( -\frac{f(j) - f(i)}{r_k} \right) \right\}. \quad (32)$$

If  $P_{ij}(r_k)$  is greater than the random number  $\delta_1$  generated within the range  $[0, 1)$ , then chromosome  $i$  is replaced by chromosome  $j$ . Neighborhood search operator is an important feature of the SA algorithm and it can enlarge the

search space with the probability of resulting in better solutions. Moreover, neighborhood search operator is one of the main operators that can increase population diversity when the algorithm is trapped into local optimal solutions.

**5.6.2. Selection.** The roulette method is adopted to select parents according to the cumulative probability, as shown in the following equation:

$$C_i = \frac{\sum_{k=1}^i f_k}{\sum_{k=1}^N f_k}, \quad (33)$$

where  $N$  represents the number of individuals in the population. A random number  $\delta_2$  is generated within  $[0, 1)$ ; if  $\delta_2 \in (C_i, C_j)$ , then chromosome  $j$  is chosen as a parent. The elitism strategy is used to reduce randomness of the algorithm. Additionally, individuals are restricted to be consecutively chosen as parents to avoid the situation when the algorithm is trapped into a local optimal solution too early.

**5.6.3. Crossover.** Two individuals are chosen as parents each time and a random number  $\delta_3$  is generated within the range  $[0, 1)$ . If  $\delta_3$  is greater than or equal to the given crossover rate  $\lambda^c$ , the crossover operator is not used and the two parents are reserved as children directly. Otherwise, the 2-opt crossover operator is performed. Figure 6 shows an example of the crossover operator where both the number of trains (i.e.,  $|L|$ ) and the number of platform tracks (i.e.,  $|I|$ ) are equal to 6.

**5.6.4. Mutation.** For each gene of a chromosome, a random number  $\delta_4$  is generated within the range  $[0, 1)$ . If  $\delta_4$  is smaller than the given mutation rate  $\lambda^m$ , then the mutation

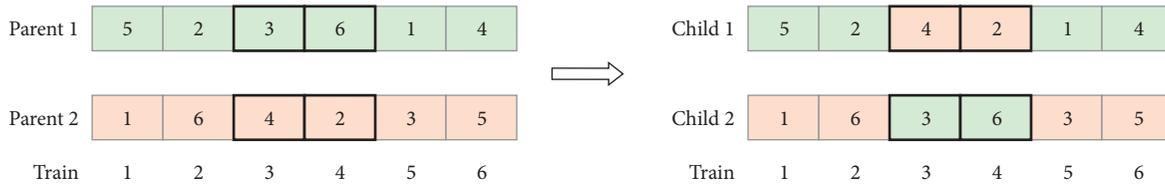


FIGURE 6: Illustration of the crossover operator.

operator is applied, i.e., a different platform track is randomly assigned to the gene.

## 6. Numerical Experiments

The proposed model is applied to a railway passenger station as shown in Figure 7, with five platform tracks (I, 3, 5, 7, 9, 11) in the downstream direction and four platform tracks (II, 4, 6, 8, 10) in the upstream direction. The time unit  $\Delta\tau$  is set to 1 min, which is practical for ensuring safe train operations at the stations [22, 25, 37]. Moreover, there are 70 downstream and upstream trains, which need to conduct the necessary operations from 16:00 to 22:00. The initial scheduled arrival and departure times of the 70 trains are available at a repository on the ResearchGate website (DOI: 10.13140/RG.2.2.18253.79849), and trains are assigned with priority values ranging from 1 to 3. The initial platform track assignment plan is listed in Table 3, and the corresponding initially scheduled train operation plan is illustrated in Figure 8. Additionally, the platform track assignment costs for the downstream and upstream trains are given in Tables 4 and 5, respectively. In particular, there is a penalty of 10,000 for trains that use the platform tracks in the opposite direction.

The train delay information is available at the time of 18:38, and it is known that 6 downstream trains and 4 upstream trains are delayed, and the estimated arrival and departure times of those delayed trains are provided in Table 6. Moreover, the 10 delayed trains in both directions are marked with the blue colour in Figure 8. Table 7 provides the parameter values used in the model. The maximum value of dwell times of all trains  $\Delta_{\max}$  equals 43 min, and the length of the planning horizon  $T$  equals 360 min. The safety time interval on the platform track  $D$  equals 6 min, and the headway between two arrival trains  $h_a$  and the headway between two departure trains  $h_d$  in the same direction are both set to 5 min. The weighting factor  $\alpha$  in the objective function is set to 200, which is determined after discussing with the train dispatchers, and the value of  $\alpha$  can be adjusted flexibly according to the preferences of the train dispatchers. Note that the value of  $\alpha$  can be flexibly adjusted by the train dispatchers according to their experiences. In addition, we believe that keeping trains on time with guaranteed train service quality is more important than assigning the trains to their preferred platform tracks, and thus the penalty parameters on train delays are relatively larger than the platform track occupancy costs. Moreover, when the values of input parameters are known, the initial values of the decision variables should be set in advance before running the GSAHA algorithm or commercial solvers. More

specifically,  $u_{l,i,1}$  is set to 1 and  $v_{l,i,1}$  is to 0 for all trains  $l \in L$  and platform tracks  $i \in I$ , such that no trains have arrived at or left the platform track at the beginning of the planning horizon. Furthermore, the variables  $w_{l,i}$ ,  $y_{l,a}$ , and  $y_{l,d}$  are set to their initial values before the train delay occurs, i.e.,  $w_{l,i} = q_{l,i}$ ,  $y_{l,a} = t_{l,a}$ , and  $y_{l,d} = t_{l,d}$  for all trains  $l \in L$  and platform tracks  $i \in I$  with the condition of  $t_{l,a} < S$ .

Based on the initial scheduled arrival and departure times as well as the input data and parameters of 70 trains in Tables 3–7, a total of 21 scenarios are generated with the number of trains ranging from 10 to 70, and 3 scenarios are generated for each fixed number of trains. For instance, three scenarios (10, 2, 0), (10, 2, 1), and (10, 2, 2) are generated when the number of trains is equal to 10. Specifically, the first number “10” in the triple (10, 2, 0) denotes the number of trains, the second number “2” in the triple (10, 2, 0) represents the number of delayed trains. Besides, the third number in a triple can take the values of 0, 1, and 2 corresponding to the three scenarios, so that the estimated arrival and departure times for delayed trains in Table 6 are randomly increased with values in the ranges of (0, 0), (−1, 1), and (−2, 2), respectively.

The MILP models for the 21 scenarios in Section 4 are solved by the commercial solver CPLEX 12.7.0 with (i.e., CPLEX-WV) and without (i.e., CPLEX-WOV) including the valid inequalities in Section 4.5. Furthermore, the maximum computation times of CPLEX-WV and CPLEX-WOV are both set to 3600 s. The GSAHA is implemented in C++ programming language. Moreover, the algorithmic parameters for GSAHA are set as follows. The number of individuals in the population  $N$  is 50, the maximum number of generations  $K$  is 300, the crossover rate  $\lambda^c$  is 0.98, the mutation rate  $\lambda^m$  is 0.1, the initial temperature  $R$  is 80,000°C, the temperature decline rate  $\eta$  is 0.9, and the temperature is increased to  $R_1 = 40,000^\circ\text{C}$  if the objective value of the best individual in the current generation remains unchanged for  $m = 5$  iterations and  $n = 3$  times of neighborhood search operations are performed each time. In addition, all of the models and algorithms are tested on a computer with Intel (R) Core (TM) i7-5600U 2.6 GHZ CPU and 12G RAM. Tables 8 and 9 list the detailed computation results of CPLEX-WOV, CPLEX-WV, and GSAHA for the 21 scenarios. More specifically, the columns “Obj 1 value (min)” and “Obj 2 value” in Table 8 represent the two components of objective values, i.e., the sum of train arrival and departure delays and total platform track assignment costs, respectively. In Table 9, the column “Obj value” denotes the overall objective values, the column “CPU time (s)” represents the computation times in sections, and the column “GAP with

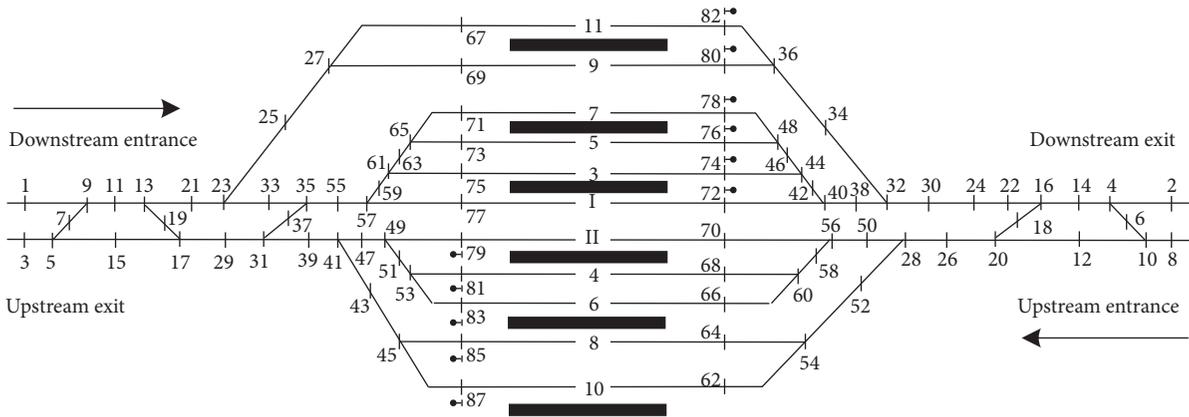


FIGURE 7: Layout of the railway passenger station.

TABLE 3: Initial platform track assignment plan between 16:00 and 22:00.

Platform track number	Occupation trains
11	$T_9, T_{29}$
9	$T_5, T_{19}, T_{31}, T_{41}, T_{49}$
7	$T_{11}, T_{21}, T_{27}, T_{33}, T_{43}, T_{47}, T_{55}, T_{63}, T_{69}$
5	$T_1, T_7, T_{15}, T_{25}, T_{35}, T_{39}, T_{53}, T_{61}, T_{67}, T_{73}$
3	$T_3, T_{13}, T_{17}, T_{23}, T_{37}, T_{45}, T_{51}, T_{57}, T_{59}, T_{65}, T_{71}, T_{75}$
I	
II	
4	$T_2, T_8, T_{18}, T_{22}, T_{30}, T_{34}, T_{40}, T_{42}, T_{48}, T_{54}, T_{60}, T_{64}$
6	$T_4, T_{12}, T_{16}, T_{24}, T_{32}, T_{38}, T_{44}, T_{50}, T_{56}, T_{62}$
8	$T_6, T_{14}, T_{20}, T_{28}, T_{36}, T_{46}, T_{52}, T_{58}$
10	$T_{10}, T_{26}$

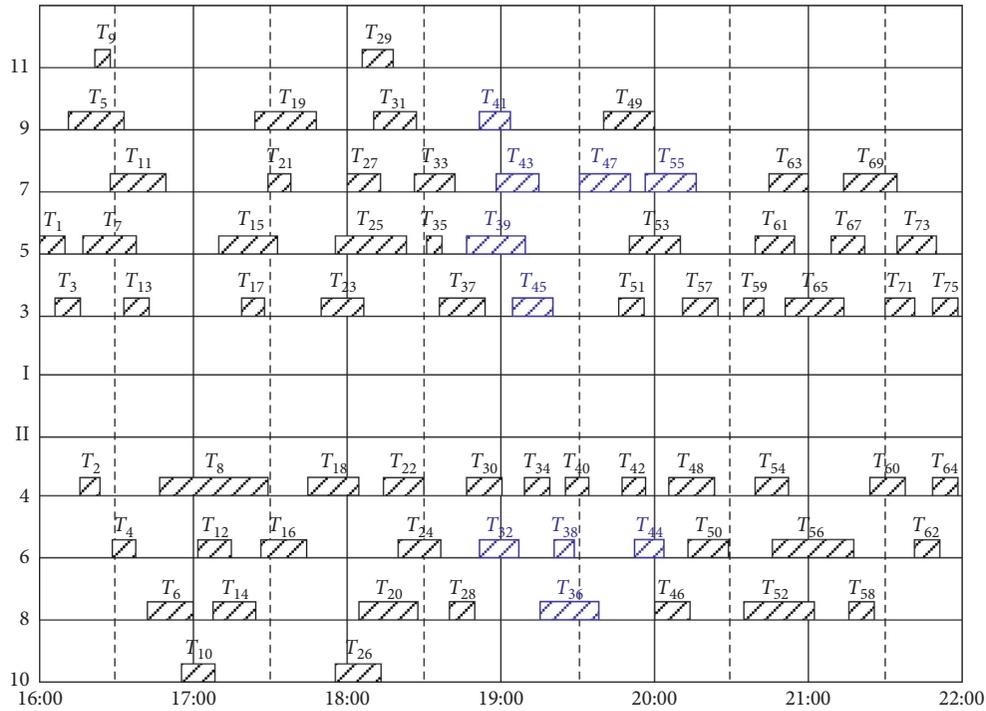


FIGURE 8: Initial platform track utilization scheme between 16:00 and 22:00.

TABLE 4: Platform track assignment costs for downstream trains with different priorities.

Train direction	Train priority	Platform track number					
		I	3	5	7	9	11
Downstream	1	600	6	12	24	48	96
	2	300	3	6	12	24	48
	3	200	2	4	8	16	32

TABLE 5: Platform track assignment costs for upstream trains with different priorities.

Train direction	Train priority	Platform track number					
		II	4	6	8	10	II
Upstream	1	600	6	12	24	48	600
	2	300	3	6	12	24	300
	3	200	2	4	8	16	200

TABLE 6: Estimated arrival and departure times for delayed trains.

Train	Arrival delay	Expected arrival time	Expected departure time	Dwell time	Priority
$T_{39}$	20	187	210	23	1
$T_{41}$	25	197	209	12	3
$T_{43}$	25	203	220	17	3
$T_{45}$	27	211	227	16	1
$T_{47}$	30	240	260	20	3
$T_{55}$	30	266	286	20	3
$T_{32}$	30	202	217	15	3
$T_{36}$	32	227	250	23	3
$T_{38}$	35	235	243	8	3
$T_{44}$	40	272	283	11	1

TABLE 7: Input parameters of the model.

Parameter	Value
Maximum value of dwell times of all trains $\Delta_{\max}$	43 min
Length of the planning horizon $T$	360 min
Safety time interval for platform track operation $D$	6 min
Sum of the length of the planning horizon and the safety time interval for platform track operation $MT$	366 min
Headway between two arrival trains $h_a$	5 min
Headway between two departure trains $h_d$	5 min
Objective function weighting factor $\alpha$	200

CPLEX-WV (%)” indicates the relative gap between the objective values of GSAHA and CPLEX-WV.

Three points can be drawn from the computation results in Tables 8 and 9. First, the computation times of CPLEX-WOV, CPLEX-WV, and GSAHA increase gradually as the number of trains increases, and the maximum computation times of those three solution methods are equal to 3600 s, 679.48 s, and 28.96 s, respectively. In particular, even if CPLEX is sped up with the valid inequalities, the computation times of GSAHA are still much fewer than that of CPLEX. Therefore, it can be shown that our proposed GSAHA can solve the reoptimization problem more efficiently in a real-time manner. Second, the objective values of GSAHA are close to those of CPLEX-WV with the maximum gap value equal to 5.66%. Third, after comparing the computation times of CPLEX-WOV and CPLEX-WV, it can be seen the valid inequalities in Section 4.5 can significantly

speed up the solving process of CPLEX. More specifically, CPLEX-WOV cannot obtain optimal solutions for the instances with the number of trains larger than or equal to 60 trains within 3600 s, which results in large objective values. In order to illustrate the feasibility of our proposed MILP model, the optimization solution of the scenario (70, 10, 0) using CPLEX-WV is illustrated in detail. Table 10 shows that the arrival and departure times of 11 trains are delayed after the reoptimization, and the new platform track assignment plan is provided in Table 11. Besides, the platform track utilization scheme is illustrated in Figure 9, and it can be seen from Figure 9 that all safety headway requirements are satisfied. Note that the 14 trains in Table 11 with bold fonts represent that those trains have been reassigned to different platform tracks, and those 14 trains are marked with the red colour in Figure 9. Besides, the convergence process of

TABLE 8: Two components of objective values of CPLEX-WOV, CPLEX-WV, and GSAHA under different scenarios.

Scenario	CPLEX-WOV		CPLEX-WV		GSAHA	
	Obj 1 value (min)	Obj 2 value	Obj 1 value (min)	Obj 2 value	Obj 1 value (min)	Obj 2 value
(10, 2, 0)	800	50	800	50	800	50
(10, 2, 1)	1200	50	1200	50	1200	50
(10, 2, 2)	800	50	800	50	800	50
(20, 3, 0)	1200	174	1200	174	1200	188
(20, 3, 1)	1800	174	1800	174	1800	174
(20, 3, 2)	1400	174	1400	174	1400	174
(30, 4, 0)	5200	278	5200	278	5200	286
(30, 4, 1)	5600	282	5600	282	5600	294
(30, 4, 2)	4400	278	4400	278	4400	283
(40, 6, 0)	800	363	800	363	800	379
(40, 6, 1)	1200	363	1200	363	1200	385
(40, 6, 2)	1200	363	1200	363	1200	395
(50, 7, 0)	8400	469	8400	469	8400	533
(50, 7, 1)	7000	469	7000	469	7000	545
(50, 7, 2)	8200	469	8200	469	8200	513
(60, 9, 0)	340400	1600	23600	583	24400	737
(60, 9, 1)	722400	2885	24200	575	25000	682
(60, 9, 2)	237800	2647	18200	579	18200	690
(70, 10, 0)	195400	1989	16400	659	16848	764
(70, 10, 1)	161600	1593	16800	685	17600	768
(70, 10, 2)	149800	2708	14600	659	15400	722

TABLE 9: Objective values and computation times of CPLEX-WOV, CPLEX-WV, and GSAHA under different scenarios.

Scenario	CPLEX-WOV		CPLEX-WV		GSAHA		
	Obj value	CPU time (s)	Obj value	CPU time (s)	Obj value	CPU time (s)	GAP with CPLEX-WV (%)
(10, 2, 0)	850	124.31	850	8.17	850	2.40	0
(10, 2, 1)	1250	51.73	1250	8.75	1250	2.43	0
(10, 2, 2)	850	67.49	850	8.23	850	2.34	0
(20, 3, 0)	1374	361.25	1374	24.66	1388	3.21	1.02
(20, 3, 1)	1974	377.20	1974	25.19	1974	3.10	0
(20, 3, 2)	1574	340.83	1574	24.02	1574	3.33	0
(30, 4, 0)	5478	326.84	5478	46.05	5486	4.75	0.15
(30, 4, 1)	5882	517.27	5882	53.25	5894	5.22	0.20
(30, 4, 2)	4678	208.92	4678	37.53	4683	5.01	0.11
(40, 6, 0)	1163	766.13	1163	46.30	1179	7.78	1.38
(40, 6, 1)	1563	480.83	1563	53.66	1585	7.83	1.41
(40, 6, 2)	1563	954.16	1563	55.27	1595	7.85	2.05
(50, 7, 0)	8869	1335.45	8869	105.22	8933	12.86	0.72
(50, 7, 1)	7469	1010.09	7469	87.42	7545	13.32	1.02
(50, 7, 2)	8669	1052.38	8669	121.58	8713	12.79	0.51
(60, 9, 0)	342000	3600	24183	661.22	25137	20.25	3.94
(60, 9, 1)	725285	3600	24775	678.17	25682	20.07	3.66
(60, 9, 2)	240447	3600	18779	604.38	18890	21.61	0.59
(70, 10, 0)	197389	3600	17059	679.48	17612	27.08	3.24
(70, 10, 1)	163193	3600	17485	344.52	18368	27.89	5.05
(70, 10, 2)	152508	3600	15259	529.75	16122	28.96	5.66

GSAHA is shown in Figure 10, where the algorithm can reach the near-optimal solution only after 70 iterations.

Meanwhile, for the scenario (70, 10, 0) in Tables 8 and 9, sensitivity analysis of different values of weighting factor  $\alpha$  is performed by increasing the value of  $\alpha$  from 0 to 10 with the step size equal to 1 for small values of  $\alpha$  and from 40 to 440 with the step size equal to 40 for relatively larger values of  $\alpha$ . Moreover, one special case is designed by setting  $\alpha = 1$  and

removing the second part of the objective function, which is aimed to simulate the case that  $\alpha$  takes an infinity value. The optimization results of CPLEX-WV and GSAHA are listed in Table 12, and the parameter settings for the GSAHA remain unchanged. In addition, we take the objective value of GSAHA for the average optimization results of 20 runs. It can be shown that the objective values of CPLEX-WV and GSAHA increase as the value of  $\alpha$  increases, and the solution

TABLE 10: Amount of secondary delay for the trains obtained by CPLEX-WV for the scenario (70, 10, 0).

Train	Priority	Secondary arrival delay (min)	Secondary departure delay (min)
$T_{39}$	1	0	4
$T_{55}$	3	0	2
$T_{32}$	3	0	3
$T_{38}$	3	2	2
$T_{40}$	3	2	2
$T_{42}$	1	5	5
$T_{46}$	2	2	2
$T_{48}$	1	2	2
$T_{50}$	1	0	1
$T_{52}$	3	2	2
$T_{54}$	1	2	2

TABLE 11: Platform tack assignment plan after the reoptimization using CPLEX-WV for the scenario (70, 10, 0).

Platform track number	Occupation trains
11	$T_9, T_{29}$
9	$T_5, T_{19}, T_{31}, T_{41}, T_{49}, T_{55}$
7	$T_{11}, T_{21}, T_{27}, T_{33}, T_{43}, T_{53}, T_{63}, T_{69}$
5	$T_1, T_7, T_{15}, T_{25}, T_{35}, T_{45}, T_{47}, T_{61}, T_{67}, T_{73}$
3	$T_3, T_{13}, T_{17}, T_{23}, T_{37}, T_{39}, T_{51}, T_{57}, T_{59}, T_{65}, T_{71}, T_{75}$
I	
II	
4	$T_2, T_8, T_{18}, T_{22}, T_{28}, T_{34}, T_{40}, T_{42}, T_{48}, T_{44}, T_{60}, T_{64}$
6	$T_4, T_{12}, T_{16}, T_{24}, T_{30}, T_{32}, T_{38}, T_{50}, T_{54}, T_{58}, T_{62}$
8	$T_6, T_{14}, T_{20}, T_{46}, T_{56}$
10	$T_{10}, T_{26}, T_{36}, T_{52}$

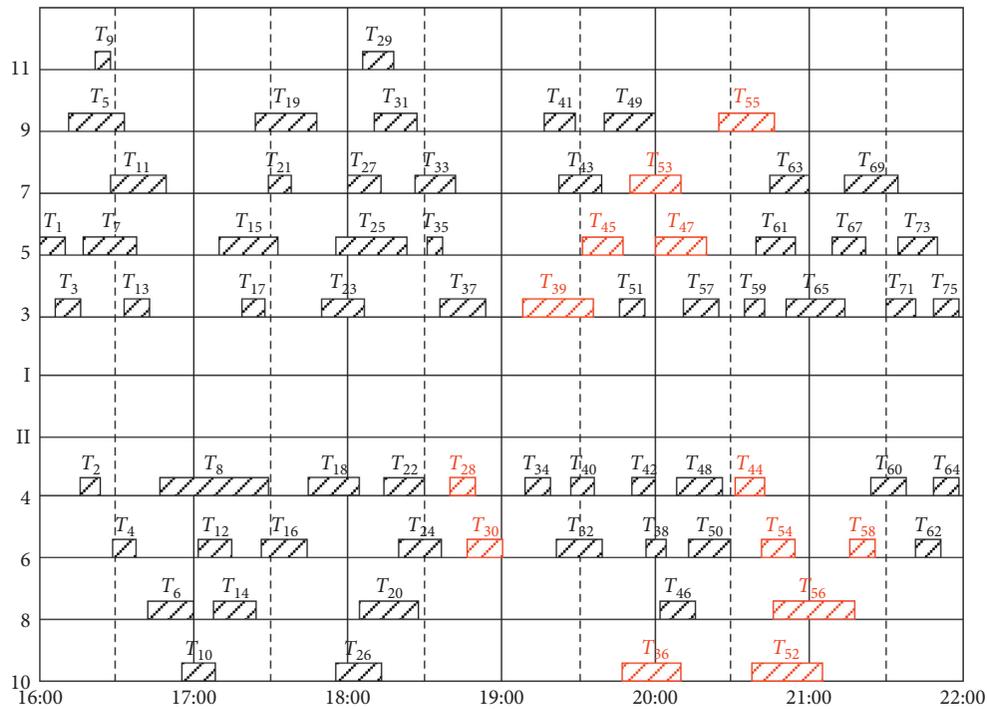


FIGURE 9: Platform track utilization scheme after the reoptimization using CPLEX-WV for the scenario (70, 10, 0).

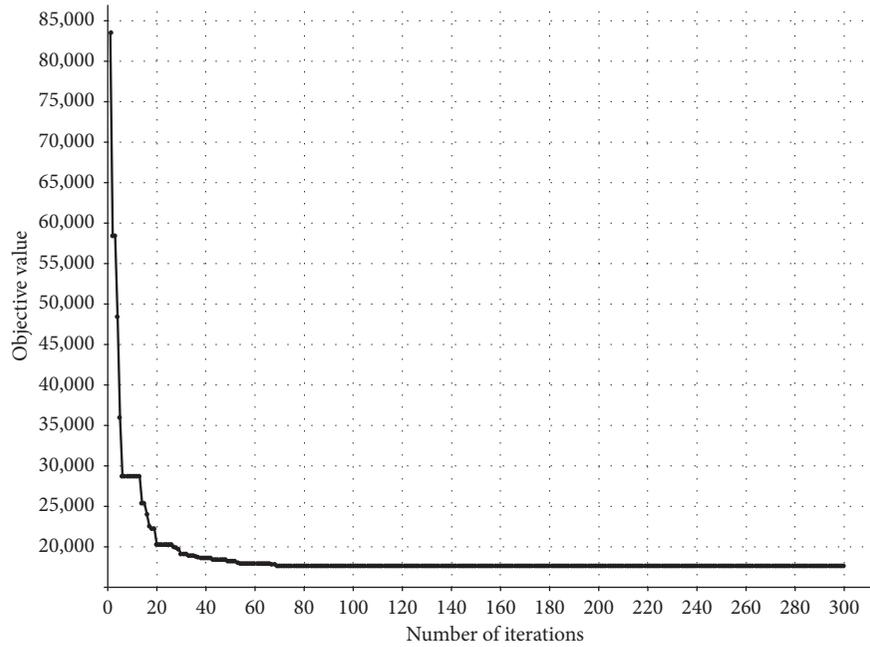


FIGURE 10: Convergence process of GSAHA for the scenario (70, 10, 0).

TABLE 12: Optimization results of CPLEX-WV and GSAHA with different values of weighting factor  $\alpha$ .

Weighting factor $\alpha$	CPLEX-WV				GSAHA				GAP with CPLEX-WV (%)
	Obj 1 value (min)	Obj 2 value	Obj value	CPU time (s)	Obj 1 value (min)	Obj 2 value	Obj value	CPU time (s)	
0	0	587	587	3600	0	477	477	32.92	-18.74
1	104	621	725	490.90	112	651	763	30.85	5.24
2	164	659	823	541	188	670	858	28.97	4.25
3	246	659	905	540.47	270	672	942	29.05	4.09
4	328	663	991	607.41	336	697	1033	29.17	4.24
5	410	663	1073	731.23	420	704	1124	29.33	4.75
6	492	659	1151	544.47	504	706	1210	29.02	5.13
7	574	659	1233	547.19	588	702	1290	28.99	4.62
8	656	663	1319	651.16	672	709	1381	29.07	4.70
9	738	669	1407	934.73	756	704	1460	29.09	3.77
10	820	663	1483	718.23	841	715	1556	28.30	4.92
40	3280	659	3939	740.22	3376	764	4140	27.60	5.10
80	6560	659	7219	589.28	6736	771	7507	28.20	3.99
120	9840	659	10499	764.37	10142	772	10914	27.52	3.95
160	13120	663	13783	447.63	13449	784	14233	27.51	3.26
200	16400	659	17059	679.48	16848	764	17612	27.08	3.24
240	19680	659	20339	388.56	20182	769	20951	27.25	3.01
280	22960	659	23619	359.88	23468	874	24342	27.49	3.06
320	26240	659	26899	595.69	26924	757	27681	27.65	2.91
360	29520	659	30179	329.28	30261	808	31069	28.23	2.95
400	32800	659	33459	340.13	33631	771	34402	28.77	2.82
440	36080	659	36739	412.67	37089	719	37808	28.22	2.91
$\infty$	82	0	82	256.36	84	0	84	30.04	2.44

times of CPLEX-WV range from 256.36 to 3600 seconds while the solution times of GSAHA only range from 27.08 to 32.92 seconds. The last column in Table 12 shows that the objective values of GSAHA are -18.74%–5.10% larger than those of CPLEX-WV. In particular, CPLEX-WV cannot obtain the optimal solution for  $\alpha = 0$  within the time limit,

while GSAHA can obtain a solution that is much smaller than that of CPLEX-WV. In addition, for the case of  $\alpha = \infty$ , it can be shown that the minimum value of the first part of the objective function is equal to 82 min, and GSAHA can also obtain an objective value of 84 min. Note that the optimality gaps of GSAHA compared with CPLEX could be

due to two reasons in particular. First, the spatial and temporal resources are allocated separately in GSAHA. Second, a heuristic method is developed to resolve the temporal conflicts between trains. Overall, the stable performance of GSAHA regarding the solution quality and solving times shows that our proposed GSAHA is suitable to serve as an effective computer-aided decision-making tool for the train dispatchers in case of train delays.

## 7. Conclusions

The problem of reoptimization of the train platforming is essential in recovering the train operations within the station while minimizing the negative impact of train delays. This paper proposes a MILP model for the reoptimization problem, where the train station is represented with the discretized platform track time-space resources. In addition, a set of valid inequalities is developed to improve the performance of the commercial solver. In addition, in order to achieve the real-time reoptimization of the train platforming problem and provide a reliable decision tool for the train dispatchers, a heuristic algorithm called GSAHA that combines the advantages of SA and GA algorithms is designed. Moreover, GSAHA first allocates the spatial resources for the trains, and the temporal conflicts between trains are further resolved through a specialized heuristic rule. To test the efficiency and effectiveness of the proposed model and algorithm, a set of real-life instances is solved by a commercial solver and GSAHA. The computational results show the proposed MILP model can effectively achieve the reoptimization of the train platforming problem, and the proposed heuristic algorithm can further speed up the solving process of the commercial solver with near-optimal solutions. In addition, the computational results also show that the performance of GSAHA is stable even when the values of weighting factor  $\alpha$  vary from 40 to 440.

The work in this paper can be extended in the following several interesting directions. First, instead of ensuring the arrival and departure safety headway between two different trains running in the same direction, the explicit consideration of train entrance and exit route conflicts can increase the station throughput capacity and reduce the train delays [10, 20]. Second, the MILP model and the heuristic algorithm GSAHA proposed in this paper can be further developed to consider different station types, such as the terminal station where trains need to perform the turnaround movement that makes the train platforming problem more complicated [42, 43]. Third, the train movements at the stations can be viewed as a set of space-time paths, and thus more efficient dual decomposition methods could be developed accordingly [7, 44, 45]. Fourth, more efficient solution algorithms can be developed for the simultaneous reoptimization of several interconnected railway stations in a railway line or even a regional railway network with more complicated structure so that the joint optimization of delay situations and train timetabling problems can be achieved [8, 19, 27, 46–49]. Finally, the integration of energy-efficient train movement with the reoptimization of train platforming can be considered for more system-wide benefits [50, 51].

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This study was supported by the National Key R&D Program (grant no. 2017YFB1200700), National Natural Science Foundation of China (grant no. U1834209), and Open Fund Project of Chongqing Key Laboratory of Traffic & Transportation (grant no. 2018TE01).

## References

- [1] National Bureau of Statistics of China, 2020, <http://data.stats.gov.cn/english/easyquery.htm?cn=C01>.
- [2] R. M. Lusby, J. Larsen, M. Ehrgott, and D. Ryan, "Railway track allocation: models and methods," *OR Spectrum*, vol. 33, no. 4, pp. 843–883, 2011.
- [3] A. A. Assad, "Models for rail transportation," *Transportation Research Part A: General*, vol. 14, no. 3, pp. 205–220, 1980.
- [4] J.-F. Cordeau, P. Toth, and D. Vigo, "A survey of optimization models for train routing and scheduling," *Transportation Science*, vol. 32, no. 4, pp. 380–404, 1998.
- [5] U. Brännlund, P. O. Lindberg, A. Nöu, and J.-E. Nilsson, "Railway timetabling using Lagrangian relaxation," *Transportation Science*, vol. 32, no. 4, pp. 358–369, 1998.
- [6] A. Caprara, M. Fischetti, and P. Toth, "Modeling and solving the train timetabling problem," *Operations Research*, vol. 50, no. 5, pp. 851–861, 2002.
- [7] Y. Zhang, Q. Peng, Y. Yao, X. Zhang, and X. Zhou, "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*, vol. 128, pp. 344–379, 2019.
- [8] Y. Zhang, A. D'Ariano, B. He, and Q. Peng, "Microscopic optimization model and algorithm for integrating train timetabling and track maintenance task scheduling," *Transportation Research Part B: Methodological*, vol. 127, pp. 237–278, 2019.
- [9] L. G. Kroon, H. Edwin Romeijn, and P. J. Zwaneveld, "Routing trains through railway stations: complexity issues," *European Journal of Operational Research*, vol. 98, no. 3, pp. 485–498, 1997.
- [10] P. J. Zwaneveld, L. G. Kroon, H. E. Romeijn et al., "Routing trains through railway stations: model formulation and algorithms," *Transportation Science*, vol. 30, no. 3, pp. 181–194, 1996.
- [11] P. J. Zwaneveld, L. G. Kroon, and S. P. M. Van Hoesel, "Routing trains through a railway station based on a node packing model," *European Journal of Operational Research*, vol. 128, no. 1, pp. 14–33, 2001.
- [12] R. Lusby, J. Larsen, D. Ryan, and M. Ehrgott, "Routing trains through railway junctions: a new set-packing approach," *Transportation Science*, vol. 45, no. 2, pp. 228–245, 2011.
- [13] R. M. Lusby, J. Larsen, M. Ehrgott, and D. M. Ryan, "A set packing inspired method for real-time junction train routing,"

- Computers & Operations Research*, vol. 40, no. 3, pp. 713–724, 2013.
- [14] V. Cacchiani, D. Huisman, M. Kidd et al., “An overview of recovery models and algorithms for real-time railway rescheduling,” *Transportation Research Part B: Methodological*, vol. 63, pp. 15–37, 2014.
- [15] L. Meng and X. Zhou, “Robust single-track train dispatching model under a dynamic and stochastic environment: a scenario-based rolling horizon solution approach,” *Transportation Research Part B: Methodological*, vol. 45, no. 7, pp. 1080–1102, 2011.
- [16] L. Meng and X. Zhou, “Simultaneous train rerouting and rescheduling on an N-track network: a model reformulation with network-based cumulative flow variables,” *Transportation Research Part B: Methodological*, vol. 67, pp. 208–234, 2014.
- [17] S. Zhan, L. G. Kroon, L. P. Veelenturf, and J. C. Wagenaar, “Real-time high-speed train rescheduling in case of a complete blockage,” *Transportation Research Part B: Methodological*, vol. 78, pp. 182–201, 2015.
- [18] S. Zhan, L. G. Kroon, J. Zhao, and Q. Peng, “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*, vol. 95, pp. 32–61, 2016.
- [19] M. Carey, “A model and strategy for train pathing with choice of lines, platforms, and routes,” *Transportation Research Part B: Methodological*, vol. 28, no. 5, pp. 333–353, 1994.
- [20] P. Chakroborty and D. Vikram, “Optimum assignment of trains to platforms under partial schedule compliance,” *Transportation Research Part B: Methodological*, vol. 42, no. 2, pp. 169–184, 2008.
- [21] A. Caprara, L. Galli, and P. Toth, “Solution of the train platforming problem,” *Transportation Science*, vol. 45, no. 2, pp. 246–257, 2011.
- [22] L. Kang, Z. Lu, and Q. Meng, “Stochastic schedule-based optimization model for track allocations in large railway stations,” *Journal of Transportation Engineering, Part A: Systems*, vol. 145, no. 3, Article ID 04019001, 2019.
- [23] D. D. L. Cardillo and N. Mione, “ $k$  L-list  $\lambda$  colouring of graphs,” *European Journal of Operational Research*, vol. 106, no. 1, pp. 160–164, 1998.
- [24] A. Billionnet, “Using integer programming to solve the train-platforming problem,” *Transportation Science*, vol. 37, no. 2, pp. 213–222, 2003.
- [25] P. Sels, P. Vansteenwegen, T. Dewilde, D. Cattrysse, B. Waquet, and A. Joubert, “The train platforming problem: the infrastructure management company perspective,” *Transportation Research Part B: Methodological*, vol. 61, pp. 55–72, 2014.
- [26] M. Carey and S. Carville, “Scheduling and platforming trains at busy complex stations,” *Transportation Research Part A: Policy and Practice*, vol. 37, no. 3, pp. 195–224, 2003.
- [27] M. Carey and I. Crawford, “Scheduling trains on a network of busy complex stations,” *Transportation Research Part B: Methodological*, vol. 41, no. 2, pp. 159–178, 2007.
- [28] L. Kang, J. Wu, and H. Sun, “Using simulated annealing in a bottleneck optimization model at railway stations,” *Journal of Transportation Engineering*, vol. 138, no. 11, pp. 1396–1402, 2012.
- [29] J. Wu, L. Kang, H. Sun, and X. Jia, “Track allocation optimization in railway station: mean-variance model and case study,” *Journal of Transportation Engineering*, vol. 139, no. 5, pp. 540–547, 2013.
- [30] M. J. Dorfman and J. Medanic, “Scheduling trains on a railway network using a discrete event model of railway traffic,” *Transportation Research Part B: Methodological*, vol. 38, no. 1, pp. 81–98, 2004.
- [31] A. D’Ariano, D. Pacciarelli, and M. Pranzo, “A branch and bound algorithm for scheduling trains in a railway network,” *European Journal of Operational Research*, vol. 183, no. 2, pp. 643–657, 2007.
- [32] J. Rodriguez, “A constraint programming model for real-time train scheduling at junctions,” *Transportation Research Part B: Methodological*, vol. 41, no. 2, pp. 231–245, 2007.
- [33] A. D’Ariano, L. Meng, G. Centulio, and F. Corman, “Integrated stochastic optimization approaches for tactical scheduling of trains and railway infrastructure maintenance,” *Computers & Industrial Engineering*, vol. 127, pp. 1315–1335, 2019.
- [34] T. M. Herrman, *Stability of timetables and train routings through station regions*, Ph.D. thesis, Swiss Federal Institute of Technology Zurich, Zürich, Switzerland, 2006.
- [35] N. Bešinović and R. M. Goverde, “Stable and robust train routing in station areas with balanced infrastructure capacity occupation,” *Public Transport*, vol. 11, no. 2, pp. 211–236, 2019.
- [36] A. D’Ariano and M. Pranzo, “An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances,” *Networks and Spatial Economics*, vol. 9, no. 1, pp. 63–84, 2009.
- [37] W. Liu, X. Zhu, and L. Kang, “Real-time track reallocation for emergency incidents at large railway stations,” *Mathematical Problems in Engineering*, vol. 2015, Article ID 296394, 11 pages, 2015.
- [38] L. Kang, S. Chen, and Q. Meng, “Bus and driver scheduling with mealtime windows for a single public bus route,” *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 145–160, 2019.
- [39] W. Xing and J. Xie, *Modern Optimization Methods*, pp. 113–147, Tsinghua University Press, Beijing, China, 2nd edition, 2006.
- [40] H. Yu, H. Fang, P. Yao, and Y. Yuan, “A combined genetic algorithm/simulated annealing algorithm for large scale system energy integration,” *Computers & Chemical Engineering*, vol. 24, no. 8, pp. 2023–2035, 2000.
- [41] H. Du, J. Fan, X. He, and M. W. Feldman, “A genetic simulated annealing algorithm to optimize the small-world network generating process,” *Complexity*, vol. 2018, Article ID 1453898, 12 pages, 2018.
- [42] Q. Zhong, R. M. Lusby, J. Larsen, Y. Zhang, and Q. Peng, “Rolling stock scheduling with maintenance requirements at the Chinese high-speed railway,” *Transportation Research Part B: Methodological*, vol. 126, pp. 24–44, 2019.
- [43] Q. Zhong, Y. Zhang, D. Wang, Q. Zhong, C. Wen, and Q. Peng, “A mixed integer linear programming model for rolling stock deadhead routing before the operation period in an urban rail transit line,” *Journal of Advanced Transportation*, vol. 2020, Article ID 3809734, 18 pages, 2020.
- [44] X. Zhou, L. Tong, M. Mahmoudi et al., “Open-source VRPLite package for vehicle routing with pickup and delivery: a path finding engine for scheduled transportation systems,” *Urban Rail Transit*, vol. 4, no. 2, pp. 68–85, 2018.
- [45] X. Chen, S. He, Y. Zhang, L. Tong, P. Shang, and X. Zhou, “Yard crane and AGV scheduling in automated container terminal: a multi-robot task allocation framework,” *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 241–271, 2020.

- [46] L. Kang, J. Wu, H. Sun, X. Zhu, and B. Wang, "A practical model for last train rescheduling with train delay in urban railway transit networks," *Omega*, vol. 50, pp. 29–42, 2015.
- [47] L. Kang, X. Zhu, H. Sun, J. Wu, Z. Gao, and B. Hu, "Last train timetabling optimization and bus bridging service management in urban railway transit networks," *Omega*, vol. 84, pp. 31–44, 2019.
- [48] L. Kang and Q. Meng, "Two-phase decomposition method for the last train departure time choice in subway networks," *Transportation Research Part B: Methodological*, vol. 104, pp. 568–582, 2017.
- [49] L. Kang, J. Wu, H. Sun, X. Zhu, and Z. Gao, "A case study on the coordination of last trains for the Beijing subway network," *Transportation Research Part B: Methodological*, vol. 72, pp. 112–127, 2015.
- [50] W. Li, Q. Peng, Q. Li, C. Wen, Y. Zhang, and J. Lessan, "Joint operating revenue and passenger travel cost optimization in urban rail transit," *Journal of Advanced Transportation*, vol. 2018, Article ID 7805168, 15 pages, 2018.
- [51] W. Li, Q. Peng, C. Wen, S. Li, X. Yan, and X. Xu, "Integrated optimization on energy saving and quality of service of urban rail transit system," *Journal of Advanced Transportation*, vol. 2020, Article ID 3474020, 22 pages, 2020.