A Heuristic Algorithm for Re-Optimization of Train Platforming in Case of Train Delays

Yongxiang Zhang [a], Qingwei Zhong [a], Chao Wen [a,b,1], Wenxin Li [a], Qiyuan Peng [a]
[a] School of Transportation and Logistics, Southwest Jiaotong University
Chengdu, 610031, China
[b] High-speed Railway Research Center, University of Waterloo
Waterloo, N2L3G1, Canada
[1] E-mail: c9wen@uwaterloo.ca, Phone: 1-2269788096

Abstract
Train platforming is critical for ensuring safety and efficiency of train operations within the stations, especially when train delays occur. This paper studies the problem of re-optimization of train platforming, where the train station is modeled using discretization of the platform track time-space resources. To solve the re-optimization problem, we propose a binary integer programming model which minimizes the weighted sum of total train delays as well as platform track utilization costs, subject to constraints defined by operational requirements. Moreover, we design an efficient heuristic algorithm to solve the model with a good precision. A real-world case is taken as an example to show the effectiveness of the proposed model and algorithm. The results show that the model established in this paper can describe re-optimization of train platforming accurately and 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.

Keywords
Train platforming; Train delay; Re-optimization; Discretization; Heuristic algorithm

1 Introduction

Train operations of the trains at stations, including arrival, dwell, and departure or passing through, are usually optimized by solving the train platforming problem (Lusby et al., 2011). In general, due to the hierarchal planning process of the railway, train timetable is specified first, and then train platforming problem is optimized with given train arrival and departure times (Lusby et al., 2011). Train platforming is a classic NP-hard combinational optimization problem (Kroon et al., 1997), and a lot of work has been done to generate high-quality train operation plan within stations. Zwaneveld et al. (1996) defined the train route as a collection of station equipment traveled by a train from inbound to the outbound of the station, and they built a mixed integer linear programming (MILP) model based on node packing problem to maximize the number of trains routed through the station. Chakroborty and Vikram (2008) 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. Caprara et al. (2011) assumed that the arrival and departure times of trains could be slightly flexible, and they presented a quadratic binary integer programming model to solve the train platforming problem. Later, Lusby et al. (2013) built MILP models based on the set packing model to maximize total revenue and minimize the total costs of all trains. Sels et al. (2014) developed a MILP model to solve the train platforming problem from strategic and tactical levels.

Trains may suffer from all kinds of disturbances and disruptions, such as bad weather, equipment failure, management factors, etc. When train delays occur, the scheduled train timetable within stations needs to be re-optimized in real time. However, very few researchers have focused on the problem of re-optimization of train platforming in case of train delays. In this study, we aim to re-optimize the train platforming 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 contributions of this study include the following three aspects. First, the train arrival and departure times and the train platform assignment are optimized simultaneously in order that the negative influence of train delays can be minimized. Second, 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 (Chakroborty and Vikram, 2008). Third, an efficient heuristic algorithm is designed to quickly obtain the near-optimal solutions for the real-time re-optimization of train platforming.

2 Analysis of platform track time-space resources

In the planned horizon $\mathscr{T}$, we handle the time resources as small time units $\Delta\tau$. The number of time units is equal to $|T| = [\mathscr{T} / \Delta\tau]$ in the entire planned 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 X,

$$X = \left[x_{i,t} \right] = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,|T|} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,|T|} \\ \cdots & \cdots & \cdots & \cdots \\ x_{|I|,1} & x_{|I|,2} & \cdots & x_{|I|,|T|} \end{bmatrix},$$

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

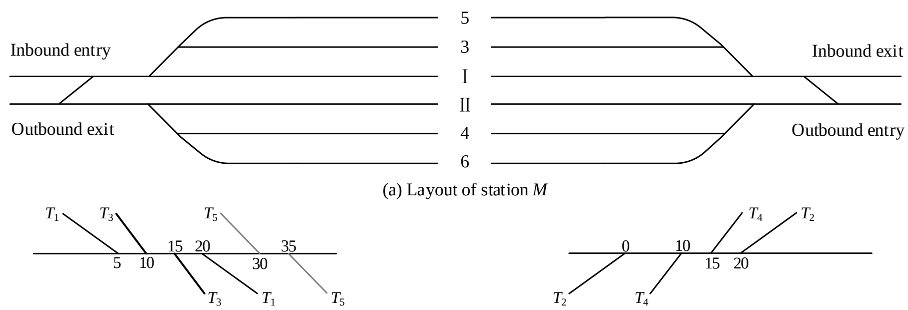
$$x_{i,t} = \begin{cases} 1, & \text{platform track time-space resource } (i,\ t) \text{ is used} \\ 0, & \text{othewise} \end{cases}$$

Fig. 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 *M* which has 6 platform tracks within the planned horizon of 60 min. The detailed schedules of train operations in both directions are given in Fig. 1 (*b*) and (*c*), and the time unit $\Delta\tau$ is set to 5 min. Platform track time-space resources and the corresponding matrix X of a feasible usage plan are described in Fig. 2 (*d*) and (*e*), respectively. The application requirements of the time-space resources modeling method for the re-optimization 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.

(3) Continuity. A train operation on one platform track with the duration equal to Δt 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., $x_{i,t} = x_{i,t+1} = x_{i,t+2} = ... = x_{i,t+\Delta t} = 1$.



(a) Layout of station *M*

(b) Train diagram of station *M* in the inbound direction

(c) Train diagram of station *M* in the outbound direction

Figure 1: Layout and train schedules of station *M*

Platform track

5	0	**T_1**	**T_1**	**T_1**	0	0	**T_5**	0	0
3	0	0	**T_3**	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0
II	0	0	0	0	0	0	0	0	0
4	0	0	**T_4**	0	0	0	0	0	0
6	**T_2**	**T_2**	**T_2**	**T_2**	0	0	0	0	0

0 5 10 15 20 25 30 35 40 45 *t*

(a) Usage of platform track time-space resources

$$
\begin{array}{c c l}
 & & \text{Platform track} \\
 & \begin{bmatrix}
0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} & \begin{matrix} 5 \\ 3 \\ \mathrm{I} \\ \mathrm{II} \\ 4 \\ 6 \end{matrix} \\
t & \begin{matrix} 5 & 10 & 15 & 20 & 25 & 30 & 35 & 40 & 45 \end{matrix} &
\end{array}
$$

(b) Platform track time-space resource matrix

Figure 2: Platform track time-space resource usage plan and time-space resource matrix

3 Modeling of re-optimization of train platforming

3.1 Parameters description

Parameters of this study are defined in Table 1. We assume that all parameters and values related to time are multiples of the time unit $\Delta\tau$.

Table 1: Illustration of Parameters

Symbol	Definition
L	Set of trains, indexed by l
L_1	Set of delayed trains
I	Set of platform tracks, indexed by i
$c_{l,i}$	Cost of train l assigned to platform track i
π_l	The 0-1 parameter equals to 1 if train l is running in the inbound direction; 0, otherwise
$q_{l,i}$	The 0-1 parameter equals to 1 if train l was initially assigned to platform track i before a delay occurs; 0, otherwise
T	Length of the planning horizon
S	The time when train delays' information is updated
$t_{l,a}$	The scheduled arrival time of train l
$t_{l,d}$	The scheduled departure time of train l
$t_{l,a}^1$	The estimated arrival time of train l when a delay occurs
$t_{l,d}^1$	The estimated departure time of train l when a delay occurs
Δ_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 dwell time among 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
α	Objective function weighting factor
M	A sufficiently large number

3.2 Variable definitions

For each train $l,k \in L$, each platform track i ($i \in I$), and each moment t ($1 \le t \le MT$), the following variables are defined in the model.

(1) Platform track choice variable $w_{l,i}$ and $z_{l,k}$

$$w_{l,i} = \begin{cases} 1, & \text{train } l \text{ chooses platform track } i \\ 0, & \text{othewise} \end{cases}$$

$$z_{l,k} = \begin{cases} 1, & \text{train } l \text{ and train } k \text{ chooses the same platform track} \\ 0, & \text{othewise} \end{cases}$$

(2) Platform track occupancy state variable $x_{l,i,t}$

$$x_{l,i,t} = \begin{cases} 1, & \text{train } l \text{ occupies platform track } i \text{ at moment } t \\ 0, & \text{othewise} \end{cases}$$

(3) Platform track occupancy state variable $u_{l,i,t}$ and clearance state variable $v_{l,i,t}$

In order to describe arrival and departure process of train l, platform track occupancy state variable $u_{l,i,t}$ as well as platform track clearance state variable $v_{l,i,t}$ are defined to denote the state of platform track i when train l arrives at and leaves from platform track i.

$$u_{l,i,t} = \begin{cases} 1, & \text{train } l \text{ has not yet arrived at platform track } i \text{ at moment } t \\ 0, & \text{othewise} \end{cases}$$

$$v_{l,i,t} = \begin{cases} 1, & \text{train } l \text{ has left platform track } i \text{ at moment } t \\ 0, & \text{othewise} \end{cases}$$

(4) Train sequence variables $\lambda_{l,k}$ and $\mu_{l,k}$

In order to describe the sequences of trains arriving at and departing from stations, the train sequence variables $\lambda_{l,k}$ and $\mu_{l,k}$ are defined as follows.

$$\lambda_{l,k} = \begin{cases} 1, & \text{train } l \text{ arrives at the station before train } k \\ 0, & \text{othewise} \end{cases}$$

$$\mu_{l,k} = \begin{cases} 1, & \text{train } l \text{ departs from the station before train } k \\ 0, & \text{othewise} \end{cases}$$

3.3 Objective function

The objective function in equation (1) contains the weighted sum of two parts. The former part is the sum of train arrival and departure delays, considering the train priority P_l and the weighting factor α, and the latter part is the total platform track occupancy costs.

$$\min z = \alpha \sum_{l \in L} P_l \left[(y_{l,a} - t_{l,a}) + (y_{l,d} - t_{l,d} - D) \right] + \sum_{l \in L} \sum_{i \in I_l} w_{l,i} c_{l,i} \tag{1}$$

3.4 Constraints

According to definitions of $x_{l,i,t}$, $u_{l,i,t}$, and $v_{l,i,t}$, the relationship among those variables can be expressed in constraint (1). Constraints (2) and (3) show that values of actual arrival time $y_{l,a}$ and actual departure time $y_{l,d}$ of train l can be inferred from $u_{l,i,t}$ and $v_{l,i,t}$. Constraint (5) requires that each train l can only be assigned to one platform track. Constraint (6) ensures that any platform track i can only be occupied by at most one train at any time t. Constraints (7)–(9) guarantee that train operations on the platform tracks should be consecutive, by enforcing the condition that the values of variables $u_{l,i,t}$ and $v_{l,i,t}$ are continuous. Constraints (10)–(15) impose the required safety headway between two arriving or departing trains running in the same direction, and the train sequence variables $\lambda_{l,k}$, $\mu_{l,k}$ as well as the platform track choice variable $w_{l,i}$ and

$z_{l,k}$ are also embedded in those constraints. Constraint (16) enforces the minimum dwell time for each train l. Note that the safety time interval for platform track operation D is included in the right 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 Δ_l of a train l is a deterministic value. Constraints (17)–(19) specify that the actual arrival and departure times of the trains should be no less than the corresponding planned arrival and departure times, respectively. Constraints (20)–(26) assign initial values to the variables $u_{l,i,t}$, $v_{l,i,t}$, $w_{l,i}$, $y_{l,a}$ and $y_{l,a}$, so that all trains adhere to their original plan before the train delay occurs. Finally, constraints (27)–(29) define the domain of variables. Note that $x_{l,i,t}$, $y_{l,a}$ and $y_{l,d}$ are intermediate variables to facilitate model definition, and their values can be inferred from $u_{l,i,t}$ and $v_{l,i,t}$.

$$x_{l,i,t} = 1-(u_{l,i,t}+v_{l,i,t}) \tag{2}$$

$$y_{l,a} = MT - \sum_{i\in I}\sum_{t=1}^{MT}(1-u_{l,i,t}) \tag{3}$$

$$y_{l,d} = MT - \sum_{i\in I}\sum_{t=1}^{MT} v_{l,i,t} \tag{4}$$

$$\sum_{i\in I} w_{l,i} = 1, \quad \forall l \in L \tag{5}$$

$$\sum_{l\in L} x_{l,i,t} \le 1, \quad \forall i \in I,\ \forall 1 \le t \le MT \tag{6}$$

$$u_{l,i,t} \ge u_{l,i,t+1} + w_{l,i} - 1, \quad \forall l \in L, \forall i \in I, \forall 1 \le t < MT \tag{7}$$

$$v_{l,i,t} \le v_{l,i,t+1} - w_{l,i} + 1, \quad \forall l \in L, \forall i \in I, \forall 1 \le t < MT \tag{8}$$

$$u_{l,i,t} \le u_{l,i,t+1} + w_{l,i}, \quad \forall l \in L, \forall i \in I, \forall 1 \le t < MT \tag{9}$$

$$y_{l,a} - y_{k,a} \ge (1-z_{l,k})h_a + z_{l,k}D - \lambda_{l,k}M, \quad \forall l,k \in L : l \ne k, \pi_l = \pi_k \tag{10}$$

$$y_{l,d} - y_{k,d} \ge (1-z_{l,k})h_d + z_{l,k}D - \mu_{l,k}M, \quad \forall l,k \in L : l \ne k, \pi_l = \pi_k \tag{11}$$

$$z_{l,k} \ge w_{l,i} + w_{k,i} - 1, \quad \forall l,k \in L, \forall i \in I_l \cap I_k : k > l, \pi_l = \pi_k \tag{12}$$

$$z_{l,k} = z_{k,l}, \quad \forall l,k \in L : k > l, \pi_l = \pi_k \tag{13}$$

$$\lambda_{l,k} + \lambda_{k,l} = 1, \quad \forall l,k \in L : k > l, \pi_l = \pi_k \tag{14}$$

$$\mu_{l,k} + \mu_{k,l} = 1, \quad \forall l,k \in L : k > l, \pi_l = \pi_k \tag{15}$$

$$\sum_{t=1}^{MT} x_{l,i,t} \ge w_{l,i} \times (\Delta_l + D), \quad \forall l \in L, \forall i \in I \tag{16}$$

$$y_{l,a} \ge t_{l,a}, \quad \forall l \in L \tag{17}$$

$$y_{l,d} \ge t_{l,d} + D, \quad \forall l \in L \tag{18}$$

$$y_{l,d} \ge y_{l,a} + \Delta_l + D, \quad \forall l \in L \tag{19}$$

$$u_{l,i,1} = 1, \quad \forall l \in L, \forall i \in I \tag{20}$$

$$v_{l,i,1} = 0, \quad \forall l \in L, \forall i \in I \tag{21}$$

$$w_{l,i} = q_{l,i}, \quad \forall l \in L, \forall i \in I : t_{l,a} < S \tag{22}$$

$$y_{l,a} = t_{l,a}, \quad \forall l \in L : t_{l,a} < S \tag{23}$$

$$y_{l,d} = t_{l,d}, \quad \forall l \in L : t_{l,a} < S \tag{24}$$

$$t_{l,a} = t^1_{l,a}, \quad \forall l \in L_1 \tag{25}$$

$$t_{l,d} = t^1_{l,d}, \quad \forall l \in L_1 \tag{26}$$

$$w_{l,i} = \{0,1\}, \quad \forall l \in L, \forall i \in I \tag{27}$$

$$u_{l,i,t}, v_{l,i,t} = \{0,1\}, \quad \forall l \in L, \forall i \in I, \forall 1 \le t \le MT \tag{28}$$

$$z_{l,k},\ \lambda_{l,k},\ \mu_{l,k} = \{0,1\}, \quad \forall l,k \in L : l \ne k, \pi_l = \pi_k \tag{29}$$

3.5 Valid equalities

Valid equalities are constraints that can strengthen the model formulation, as shown in constraints (30)–(33).

$$u_{l,i,t} \ge 1 - w_{l,i}, \quad \forall l \in L, \forall i \in I, \forall 1 \le t \le MT \tag{30}$$

$$v_{l,i,t} \le w_{l,i}, \quad \forall l \in L, \forall i \in I, \forall 1 \le t \le MT \tag{31}$$

$$x_{l,i,t} \le w_{l,i}, \quad \forall l \in L, \forall i \in I, \forall 1 \le t \le MT \tag{32}$$

$$x_{l,i,t} = 0, \quad \forall l \in L, \forall i \in I, \forall t < t_{l,a} \text{ or } t > t_{l,d} + \Delta_{\max} + D \tag{33}$$

Principle of valid inequalities (30), (31) and (32) are similar. For example, in valid inequality (30), if train l occupies platform track i, then valid inequality (30) is equivalent to $u_{l,i,t} \ge 0$ which turns out to be ineffective. However, if train l does not occupy the platform track i, then valid inequality (30) is equivalent to $u_{l,i,t} \ge 1$ which implies $u_{l,i,t} = 1$. Valid inequality (33) 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 $t < t_{l,a}$ or $t > t_{l,d} + \Delta_{\max} + D$.

4 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 optimization model efficiently and effectively (Xing et al., 2006). The GSAHA algorithm combines the advantages of genetic algorithm (GA) and simulated annealing algorithm (SA). Moreover, GSAHA is robust on the convergence performance while avoiding being trapped into the local optimal solutions. The implementation details for the components of GSAHA are illustrated as follows.

4.1 Chromosome representation

Fig. 3 shows the one-dimensional real-value encoding method that is used to represent chromosomes. Each chromosome denotes a platform track assignment plan, i.e., if the value of the l^{th} gene is equal to i, then the l^{th} train is assigned to platform track i with its scheduled arrival and departure time. The length of each chromosome is equal to the number of trains $|L|$, and the genes in a chromosome are numbered in decreasing order according to the scheduled arrival time of trains, where the value range of each gene is located within the range $[1, |\mathrm{I}|]$, and there could be $|I|^{|L|}$ chromosomes in total.

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

Figure 3: Illustration of chromosome representation

4.2 Generate 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 number is 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 rest $|L| - \lfloor |L| / (|I|-1) \rfloor \cdot (|I|-1)$ 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.

4.3 Obtain a feasible solution

The chromosome designed in Section 4.1 only assigns trains to 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, namely, the headway 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 in Section 3.4:

Step 1. Sort all trains in decreasing order by their scheduled or estimated arrival time and number them from 1 to $|L|$.

Step 2. 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.

Algorithm 1: heuristic method to resolve the temporal conflicts with given train order

For each train i $(1 \le i \le |L|)$
    For each train j $(1 \le j < i)$
        If train i conflicts with train j

Fix the arrival and departure times of train i and record the weighted-sum delay amount ϑ_i after resolving the conflicts of trains number before train i;
Fix the arrival and departure times of train j and record the weighted-sum delay amount ϑ_j after resolving the conflicts of trains number before train i;
If $\vartheta_i \le \vartheta_i$
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 $\vartheta_i \le \vartheta_i$
End If train i conflicts with train j
End For each train j $(1 \le j < i)$
End For each train i $(1 \le i \le |L|)$

Step 3. Calculate the weighted sum of arrival and departure delays compared to the scheduled or estimated arrival and departure times. This operation considers all trains in set L and the platform track occupancy costs. The value calculated during this step serves as the objective value of the corresponding chromosome.

4.4 Fitness function

The fitness function in equation (34) is designed to evaluate each individual such that the algorithm can achieve a better convergence performance:

$$f_i(t_k) = \exp\left\{-\frac{f(i) - f_{\min}}{t_k}\right\} \quad , \tag{34}$$

where t_k represents the temperature at the k^{th} generation, $f(i)$ represents the objective value of the i^{th} chromosome, $f_{\min}$ represents the minimal objective value at the k^{th} generation, and $f_i(t_k)$ represents fitness value of the i^{th} chromosome when the temperature is t_k. Fitness function in equation (34) is an important feature of the simulated annealing (SA) algorithm, and it has a good capacity to accelerate the convergence of the algorithm.

4.5 Temperature decline function

After determining the initial temperature T, the temperature decline function in equation (35) is used to lower the temperature at each iteration:

$$t_k = T \cdot \alpha^k \quad , \tag{35}$$

where t_k represents the temperature at the k^{th} generation, and the constant α

represents the temperature decline rate in the SA algorithm.

4.6 Genetic operators

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 $P_{ij}(t_k)$ in equation (36).

$$P_{ij}(t_k) = \min\left\{1, \exp\left(-\frac{f(j)-f(i)}{t_k}\right)\right\} \tag{36}$$

If $P_{ij}(t_k)$ is greater than the random number r_1 generated within the range [0, 1), then chromosome i is replaced by chromosome j. Neighborhood search operator is another 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.

Selection

Roulette method is adopted to select parents according to the cumulative probability, as shown in equation (37):

$$C_i = \sum_{k=1}^{i} f_k \Big/ \sum_{k=1}^{N} f_k \quad , \tag{37}$$

where N represents the number of individuals in the population. A random number r_2 is generated within [0, 1), if $r_2 \in (C_i,\ C_j)$, 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.

Crossover

Two individuals are chosen as parents each time and a random number r_3 is generated within the range [0, 1). If r_3 is greater than or equal to the given crossover rate, then the crossover operator is not used and the two parents are reserved as children directly; otherwise, 2-point crossover operator is performed.

Mutation

For each gene of a chromosome, a random number r_4 is generated within the range [0, 1). If r_4 is smaller than the given mutation rate, then the mutation operator is applied, i.e., a different platform track is randomly assigned to the gene.

5 Numerical experiments

The proposed model is applied to a railway passenger station as shown in Fig. 4, with five platform tracks (I, 3, 5, 7, 9, 11) in the inbound direction, and four platform tracks (II, 4, 6, 8, 10) in the outbound direction. The time unit $\Delta\tau$ is set as 1 min. There are 70 inbound and outbound trains which need to conduct the necessary operations from 16:00 to 22:00. Trains have assigned priorities from 1 to 3, and the initially scheduled train operation plan within the station is illustrated as shown in Table 2 and Fig. 5. Additionally, the platform track occupancy costs for the inbound and outbound trains are given in Tables 3 and 4. There is a penalty of 10,000 for trains which use the platform tracks in the opposite direction. Moreover, it is known that 6 inbound trains and 4 outbound trains are delayed at 18:38, and the estimated arrival and departure times of those delayed trains are given in Table 5. The maximum dwell time $\Delta_{\max}$ is 43 min, and the length of the scheduled horizon T is 360 min. The safety interval time on the platform track D is 6 min, and the headway between two arriving or departing trains in the same direction are set as 5 min. The weighting factor α is set as 200. Please note that the value of α can be flexible adjusted by the train dispatchers. 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.

First, we use the commercial solver CPLEX 12.7.0 to solve the model in section 2. The test computer is an Intel(R) Core(TM) i7-5600U 2.6GHZ CPU with 12G RAM. CPLEX can obtain optimal solutions after *608 seconds* with the *objective value of 17,059*. Table 7 shows that arrival and departure times of 11 trains are delayed after the re-optimization, and 14 trains are assigned different platform tracks. The new train operation plan within the station is shown in Table 6 and Fig. 6, where all safety headway requirements are satisfied. Please note that the trains in Table 6 with bold fonts represent that those trains have been reassigned to different platform tracks.

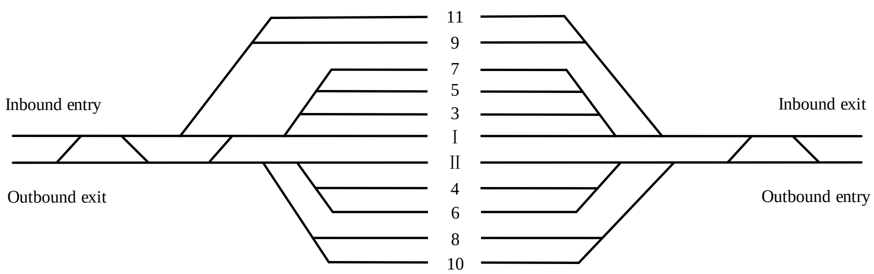


Figure 4: Layout of the railway passenger station

Table 2: Initial platform tack 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}

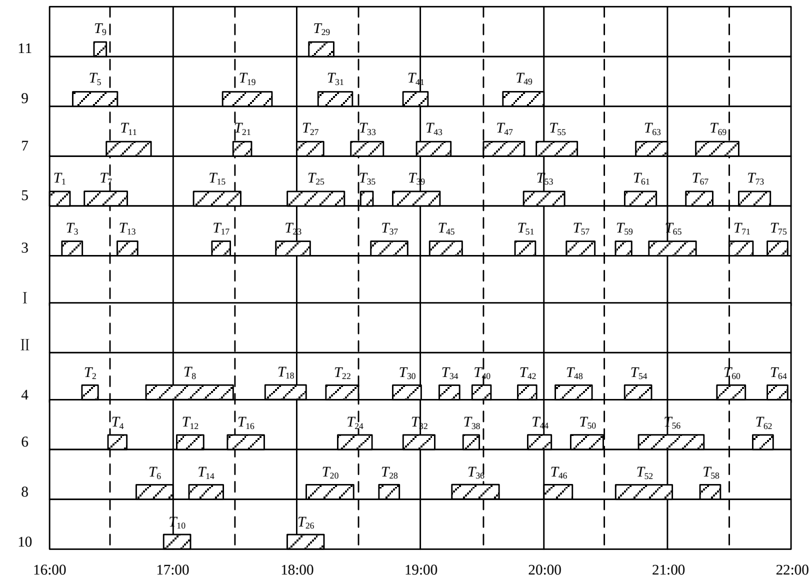


Figure 5: Arrival and departure track utilization scheme between 16:00 and 22:00

Table 3: Platform track occupancy costs for inbound trains with different priorities

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

Table 4: Platform track occupancy costs for outbound trains with different priorities

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

Table 5: 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 6: Platform tack assignment plan after re-optimization with CPLEX

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

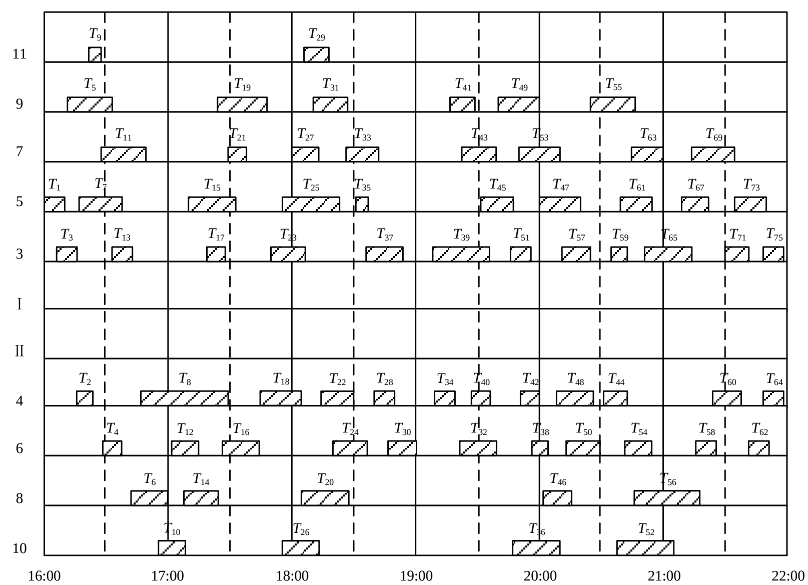


Figure 6: Platform track use scheme after re-optimization with CPLEX

Table 7: Amount of secondary delay for the trains obtained by CPLEX

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

Parameters for GSAHA are set as follows. The number of individuals in the population is 50, the maximum number of generations is 300, the crossover rate is 0.98, the mutation rate is 0.1, the initial temperature T is 8000 ℃, the temperature decline rate α is 0.9, and the temperature is increased to 4000 ℃ if the objective value of the best individual in the current generation remains unchanged for 3 iterations. The GSAHA is implemented in C++, and the average objective value of GSAHA for total 20 runs is *17,612*, which is only 3.24% higher than the optimal solution of CPLEX. In addition, the average running time of the GSAHA is only *27 seconds*. The convergence process of the simulated annealing hybrid algorithm for a specific run is shown in Fig. 7, where the algorithm can reach the near-optimal solution only after 70 iterations.

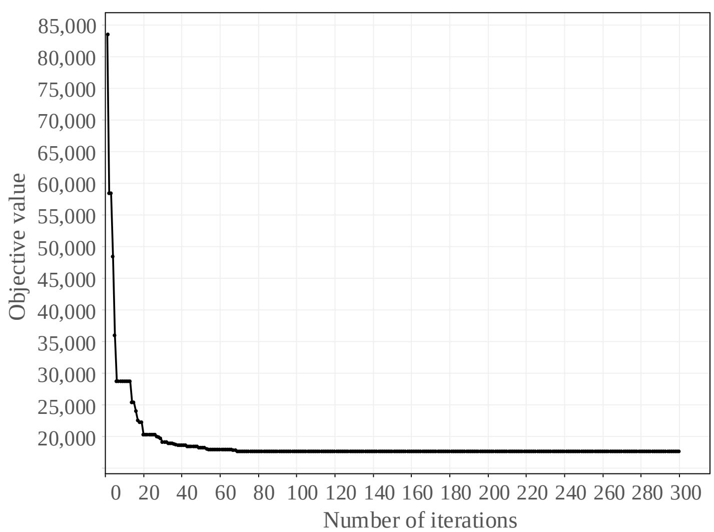


Figure 7: Convergence process of GSAHA

Meanwhile, the sensitivity analysis of different values of weighting factor α is performed by increasing the value of α from 40 to 440 with the step size equal to 40. The optimization results of CPLEX and GSAHA are listed in Table 8, and the parameter settings for the GSAHA remain unchanged, and the objective value of GSAHA takes the average results of 20 times. It can be shown that the objective values of CPLEX and GSAHA increase as the value of α increases, and the solution times of CPLEX range from *329 to 764 seconds* while the solution times of GSAHA only range from *27 to 29 seconds*. In addition, the objective values of GSAHA are *2.80%–5.10%* larger than that of CPLEX.

Hence, the stable performance of GSAHA regarding the solution quality and solution times show 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.

Table 8: Optimization results of CPLEX and the GSAHA with different weighting factors

Weighting factor α	**CPLEX**		**GSAHA**		
	Objective value	**CPU time (sec)**	**Objective value**	**CPU time (sec)**	**GAP with CPLEX (%)**
40	3939	740	4140	28	5.10
80	7219	589	7507	28	3.99
120	10499	764	10914	28	3.95
160	13783	447	14233	28	3.26
200	17059	679	17612	27	3.24
240	20339	388	20951	27	3.01
280	23619	360	24342	27	3.06
320	26899	596	27681	28	2.91
360	30179	329	31069	28	2.95
400	33459	340	34402	29	2.82
440	36739	412	37808	28	2.91

6 Conclusions

The problem of re-optimization of the train platforming is essential in recovering the train operations within the station and minimizing the negative influences of train delays. This paper proposes a MILP re-optimization model, where the train station is represented using discretized platform track time-space resources. The resulting model is solved by CPLEX and the designed heuristic algorithm GSAHA. The effectiveness of the proposed MILP model is verified by using the CPLEX solver, and the proposed heuristic algorithm further speeds up the solving process with near-optimal solutions. In addition, the performance of GSAHA is stable when the values of weighting factor α vary from 40 to 440.

The work in this paper can be extended in several interesting directions. First, instead of ensuring the arrival and departure safety headway between two different trains (Chakroborty and Vikram, 2008), the explicit consideration of train entrance and exit route conflicts can increase the station throughput capacity and reduce the train delays (Zwaneveld et al., 1996). 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 turn-around movement which makes the train platforming problem more complicated. Third, the effectiveness of the heuristic algorithm GSAHA can be tested and improved for bigger railway stations with more complex station layout structure.

Acknowledgements

This work is supported by the National Key R&D Program [grant no. 2017YFB1200700] and National Nature Science Foundation of China [grant no. 71871188, grant no.

U1834209]. The first and second authors gratefully acknowledges the financial support from the China Scholarship Council [201707000080, 201707000041].

References

Caprara, A., L. Galli, P. Toth., 2011. "Solution of the train platforming problem". *Transportation Science*, Vol. 45, No. 2, pp. 246-257.

Chakroborty, P., Vikram, D., 2008. "Optimum assignment of trains to platforms under partial schedule compliance". *Transportation Research Part B: Methodological*, 42(2), 169-184.

Kroon, L. G., H. Edwin Romeijn, P. J. Zwaneveld., 1997. "Routing trains through railway stations: complexity issues". *European Journal of Operational Research*, Vol. 98, No. 3, pp. 485-498.

Lusby, R. M., J. Larsen, M.E. David Ryan., 2011. "Railway track allocation: models and methods". *OR spectrum*, Vol. 33, No. 4, pp. 843-883.

Lusby, R., J. Larsen, M. Ehrgott., 2013. "A set packing inspired method for real-time junction train routing". *Computers & Operations Research*, Vol. 40, No. 3, pp. 713-724.

Sels, P., P. Vansteenwegen, and T. Dewilde. The train platforming problem: the infrastructure management company perspective. *Transportation Research Part B: Methodological*, Vol. 61, 2014, pp. 55-72.

Xing, W., Xie, J., 2006. "Modern optimization methods (2nd Edition)". Beijing: Tsinghua University Press, 113-147.

Zwaneveld, P. J., L. G. Kroon, H. Edwin Romeijn, 1996. "Routing trains through railway stations: model formulation and algorithms". *Transportation Science*, Vol. 30, No. 3, pp. 181-194.