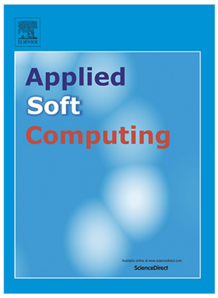
# Journal Pre-proof




An efficient evolutionary algorithm for high-speed train rescheduling under a partial station blockage

Rongsheng Wang, Qi Zhang, Xuewu Dai, Zhiming Yuan, Tao Zhang, Shuxin Ding, Yaochu Jin

PII: S1568-4946(23)00608-7
DOI: https://doi.org/10.1016/j.asoc.2023.110590
Reference: ASOC 110590

To appear in: *Applied Soft Computing*

Received date : 29 November 2022
Revised date : 27 April 2023
Accepted date : 22 June 2023

Please cite this article as: R. Wang, Q. Zhang, X. Dai et al., An efficient evolutionary algorithm for high-speed train rescheduling under a partial station blockage, *Applied Soft Computing* (2023), doi: https://doi.org/10.1016/j.asoc.2023.110590.

# An efficient evolutionary algorithm for high-speed train rescheduling under a partial station blockage

Rongsheng Wang[a,b], Qi Zhang[c,d,*], Xuewu Dai[e], Zhiming Yuan[c,d], Tao Zhang[c,d], Shuxin Ding[c,d], Yaochu Jin[f,g]

[a] *Scientific and Technological Information Research Institute, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China*
[b] *Office of Scientific and Technological Achievements and Intellectual Propert, China State Railway Group Corporation Limited, Beijing 100081, China*
[c] *Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China*
[d] *Train Operation Control Laboratory for High-Speed Railway, National Engineering Research Center of System Technology for High-Speed Railway and Urban Rail Transit, Beijing 100081, China*
[e] *State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China*
[f] *Faculty of Technology, Bielefeld University, D-33615 Bielefeld, Germany*
[g] *Department of Computer Science, University of Surrey, Guildford GU2 7XH, U.K.*

**Abstract**

This paper investigates the high-speed train rescheduling (HSTR) problem under a partial station blockage and proposes an efficient problem-specific strengthen elitist genetic algorithm (PS-SEGA) for HSTR. Firstly, a HSTR model subject to train operation constraints is established to minimize the total train delay. A permutation-based encoding method is developed to define an efficient search space based on the train departure sequence. A heuristic decoding method is employed to eliminate all train operation constraints and output the rescheduled timetable. Moreover, a hybrid initialization method involving an efficient heuristic strategy (EHS) is put forward to accelerate the convergence speed of PS-SEGA. Using problem-specific knowledge, EHS generates an efficient and feasible solution for the initial population. Finally, a restart strategy is presented to maintain genetic diversity. Compared with other advanced evolutionary algorithms and their improved variants also using the improvements of PS-SEGA, experimental results demonstrate the effectiveness of the proposed PS-SEGA for addressing HSTR scenarios under the partial station blockage. As for the scenarios that CPLEX cannot obtain optimal solutions within 10 minutes, PS-SEGA can provide quasi-optimal solutions in real time. Furthermore, compared with the other two heuristics algorithms (i.e., First-Scheduled-First-Served and EHS), PS-SEGA can give the train departure sequence with a smaller total train delay.

*Keywords:* High-speed railways, Train rescheduling, Evolutionary algorithm, Genetic algorithm, Permutation-based optimization.

## 1. Introduction

High-speed railways have become the preferred way of public transportation with an environmentally friendly, high punctuality, and comfortable service. Some countries, e.g., Germany, Japan, and China, gradually build high-speed railways to strengthen city links. For instance, more than 42000 kilometers of high-speed railways have been completed in China until 2022. High-speed railways are significant in promoting economic construction, speeding up carbon neutrality, and improving coordinated urban development.

*Corresponding author at: Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China.

*Email addresses:* wrs20138437@126.com (Rongsheng Wang), gorgeous@139.com (Qi Zhang), daixuewu@mail.neu.edu.cn (Xuewu Dai), yuanzhm@rails.cn (Zhiming Yuan), 13701193534@139.com (Tao Zhang), dingshuxin@rails.cn (Shuxin Ding), yaochu.jin@uni-bielefeld.de (Yaochu Jin)

With the growth of operating mileage and the increase in passenger flow, high-speed railway companies should ensure operation efficiency and drive more passengers to their destinations. The safe and efficiency of railway management are guaranteed by the train planned timetable that stipulates the arrival or departure times and departure sequences. However, unexpected emergencies (e.g., strong winds, railway infrastructure malfunctioning, earthquakes, etc.) often happen. Dispatchers require to adjust the train timetable to restore the original order. According to the time and position of the emergencies, dispatchers will install a segment blockage, a station blockage, or a temporary speed restriction. Influenced by emergencies, the arrival or departure times of trains may deviate from the planned timetable. It is urgent to draw up available contingency plans for each emergency. The above process can be called the problem of high-speed train rescheduling (HSTR). Nowadays, HSTR is still mainly handled by dispatchers based on their experience in emergencies. It is difficult to guarantee both the optimality and rapidity of the HSTR solution. Therefore, it is imperative to investigate an approach that can obtain a quasi-optimal HSTR solution in real time.

As an NP-hard problem [1, 2], HSTR is to obtain a new timetable when the original one becomes infeasible by adjusting train arrival and departure times, selecting proper routes at stations, or canceling trains [3]. The computation time of solving HSTR appears to be exponential with the growth of trains and stations. The approaches to addressing HSTR comprise the following three types.

**Operational Research (OR)**: Since the problem of optimal train scheduling was proposed by Szpigel in 1973 [4], much research has concentrated more on OR. The common models of OR approach involve the mixed integer linear programming (MILP) model, job shop scheduling (JSP) model, and alternative graph model. The MILP model was usually established based on train operation constraints in block sections and stations to minimize the train delay [5, 6], the number of canceled trains [7–9], or the passenger travel time [10]. Mascis and Pacciarelli first proposed an alternative graph model for the JSP problem [11]. Then D'Ariano and Corman *et al.* [12–14] developed the alternative graph model in the real-time railway management system to resolve conflicts in block sections and stations. In addition, the alternative graph model in HSTR was usually associated with the JSP model [15], where a train corresponds to a job and a block section to a machine.

**Evolutionary Algorithms (EAs)**: Utilizing the characteristics of genetic crossover and variation in biological populations, EAs have resolved many HSTR problems in recent years, e.g., genetic algorithm [16–19], NSGA-II [20], ant colony optimization [21–23], etc. Some encoding/decoding methods, heuristic initialization methods, and restart strategies were used to accelerate the convergence of EAs in some NP-hard problems [24–28]. Although it is difficult to guarantee the optimality of each solution, EAs can provide a quasi-optimal solution at less computation cost. With appropriate improvement strategies, EAs are suitable to address the HSTR problem in real time.

**Reinforcement learning (RL)**: With the rapid development of intelligent technologies, artificial intelligence, represented by reinforcement learning (RL), has been gradually applied in theoretical research and engineering practice. In railway optimization problems, RL has been widely used in automatic metro train control [29], passenger inflow control [30], energy storage system for urban rail [31], heavy haul train control [32], etc. In particular, much research has focused on the train rescheduling problem using RL on a single-track railway [33, 34], which mainly addressed the problem of deadlock resolution and train delay minimization. Other RL approaches, e.g., Deep Q Network [35], Monte Carlo tree search [36], have also been associated with RL to resolve HSTR.

OR has advantages in providing an optimal solution for HSTR, but branch and bound rules need to be designed to improve the computation efficiency. Unlike OR, the offline training model generated by RL can directly provide an online HSTR solution in real time [35]. In our previous study, we proposed an approach called Monte Carlo tree search in a RL environment to adjust arrival and departure times to minimize train delay [36]. The difficulties of RL for resolving HSTR are how to design a RL environment and search for an optimal solution during the training. Compared with OR and RL, EAs do not always obtain an optimal solution but can provide an available and quasi-optimal solution in real time using problem-specific knowledge. In summary, if the HSTR approach requires to be applied in an existing system, it is unsuitable to pursue only the solution optimality, ignoring the real-time performance. In particular, EAs can maintain a trade-off between the optimality and rapidity of the HSTR solution.

Recently, few studies have solved HSTR by combining EAs with the characteristics of the problem. Therefore, we propose an efficient problem-specific strengthen elitist genetic algorithm (PS-SEGA) for HSTR under a partial station blockage to minimize the total train delay by reordering and retiming trains simultaneously. More significantly, we maintain a trade-off between the optimality and rapidity of the HSTR solution. Aiming at the limitations of traditional EAs addressing HSTR, we conclude the contributions of this paper as follows.

(1) The encoding method is used to input decision variables to PS-SEGA and determines the population size. The traditional integer-based encoding method uses train arrival and departure times to denote the population individuals of PS-SEGA, but it consumes too much time in solving HSTR. To improve search efficiency, we present a novel permutation-based encoding method to denote the train departure sequence at the blockage station as the population individual. There is no need to consider all the arrival and departure times in the train timetable. This encoding method can avoid invalid searches and greatly improve solution efficiency.

(2) Though the existing decoding method with integer-based encoding can directly output the rescheduled timetable, it needs to handle all train operation constraints and consumes too much time. To improve the computation efficiency, we propose a heuristic decoding method to realize no-constraint processing. The train departure sequence calculated by the proposed permutation-based encoding method is converted into the final rescheduled timetable.

(3) The traditional rand initialization method provides different individuals for the initial population to keep good diversity. However, this method leads to low search efficiency. To accelerate the convergence speed, we develop a hybrid initialization method for PS-SEGA to provide an efficient solution for the initial population. This population involves random individuals and an *efficient heuristic* strategy that utilizes problem-specific knowledge of the HSTR problem.

The remainder of the paper is formulated as follows. Section 2 establishes a mathematical model of HSTR. Section 3 presents the improvements of the proposed PS-SEGA for HSTR. Numerical experiments and algorithm comparisons are performed to validate the performance of PS-SEGA in Section 4. Section 5 summarizes the paper and provides further work.

## 2. Model formulation

This paper considers a double-track high-speed railway line containing $G$ trains and $I$ stations. Train $g$ departs from the origin station, passes or stops at the intermediate station $i$, and arrives at the terminus $I$. Assume that a partial station blockage occurs at station $i'$. The scenario under the partial station blockage considered in this paper can be illustrated in Fig. 1, where $t_{\text{start}}$ and $t_{\text{end}}$ represent the start and end time instants of the blockage, respectively. Notably, trains can only arrive at the station but cannot depart in the station-blocked area. Facing the station blockage in the HSTR problem, dispatchers should determine the departure sequence of all influenced trains to minimize the total train delay. However, this problem is addressed by the dispatcher's experience and is not easily solved in real time. Therefore, we propose an algorithm called PS-SEGA to reorder trains at the blockage station $i'$ to minimize the total train delay. To better illustrate the model, we lists all the parameters in Table 1. The assumptions of HSTR are as follows.

(1) Trains cannot arrive at the next station or depart from the current station earlier than the planned time.

(2) Determined by dispatchers, the start and end time instants ($t_{\text{start}}$ and $t_{\text{end}}$) of the partial station blockage are known and fixed when rescheduling trains.

(3) The strategy of reordering trains is merely permitted at station $i'$.

(4) Trains run according to the minimum dwell times, headways, running times, and tracing times in each block section.

Assumption 1) ensures the necessary service for alighting and boarding passengers at all stations. Assumption 2) depicts the regulations that trains must comply with under a partial station blockage. The installed partial station blockage does not affect the trains leaving before $t_{\text{start}}$. However, the trains in blocked window $[t_{\text{start}}, t_{\text{end}})$ are only allowed to depart after $t_{\text{end}}$. Assumption 3) is to meet the actual high-speed railway conditions in China according to the technical specification in the centralized traffic control system [37]. The technical specification regulates that trains are rarely reordered along the railway line except for the blockage station. Following Assumption 4), trains

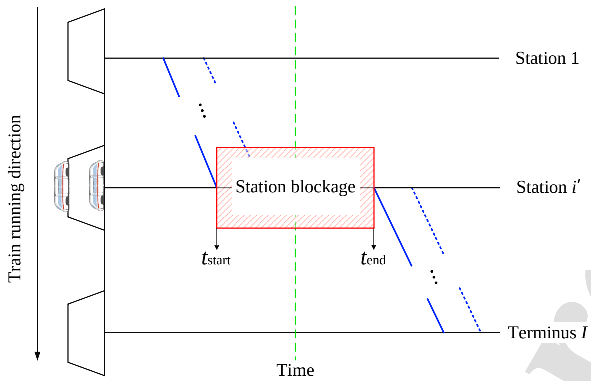



Figure 1. Illustration of the partial station blockage in time (horizontal axis) and space (vertical axis).

should compress interval buffer times by tracking each other closely under a quasi-moving block system to minimize the total train delay.

**Remark 1:** Note that we do not consider a segment blockage. The reasons are as follows. The segment blockage between two stations only causes all trains to arrive late at the next station. The strategy of reordering trains is not allowed at the subsequent stations, according to Assumption 3). Therefore, trains run only following the original order in the planned timetable. The train departure sequence hardly requires to be changed under the segment blockage. Similarly, we do not consider a complete station blockage under which the operation of receiving and sending trains is failed. This blockage causes all trains to stop in front of the blockage station. When the blockage ends, the trains pass the blockage station following the departure sequence in the planned timetable. There is no need to reorder trains.

### 2.1. Objective function

When a partial station blockage causes train delays, railway companies usually focus on recovering trains in regular operating order. Hence, following Assumption 1), the objective function $Y$ of HSTR is defined as the total train delay, i.e.,

$$Y = \sum_{g=1}^{G} \sum_{i=1}^{I} \left[ \left( a_{g,i} - \bar{a}_{g,i} \right) + \left( d_{g,i} - \bar{d}_{g,i} \right) \right] \tag{1}$$

where $a_{g,i}$ and $d_{g,i}$ are the actual arrival and departure times for train $g$ ($g \in \{1, 2, ..., G\}$) at station $i$ ($i \in \{1, 2, ..., I\}$), respectively. The planned arrival time $\bar{a}_{g,i}$ and departure time $\bar{d}_{g,i}$ are obtained from the planned timetable.

### 2.2. Constraints

#### 2.2.1. Constraints of the partial station blockage

Note that we consider reordering trains at the blockage station $i'$. When the blockage ends, assume that train $g'$ is the first train departing from station $i'$. Let $\bar{g}$ and $g''$ represent the train not affected by the blockage and the train influenced by the blockage respectively. Followed from Assumptions 2) – 4), the actual departure times for trains are given by

$$\begin{cases} d_{\bar{g},i'} = \bar{d}_{\bar{g},i'}, \ \bar{d}_{\bar{g},i'} < t_{\text{start}}, \ \bar{g} \in M_{\text{no}} \\ d_{g',i'} = t_{\text{end}}, \ \bar{d}_{g',i'} > t_{\text{start}} \\ d_{g'',i'} = \bar{d}_{g'',i'} + h^{\text{dd}}_{g''-1,g'',i'}, \ \bar{d}_{g'',i'} > t_{\text{start}} \ , g'' \in M_{\text{blk}}. \end{cases} \tag{2}$$

In Eq. (2), note that $M_{\text{no}} = \{1, 2, ..., g' - 1\}$ is the set of the trains that are not influenced by the partial station blockage. If the planned departure time $\bar{d}_{\bar{g},i'}$ for train $\bar{g}$ at station $i'$ is earlier than $t_{\text{start}}$, train $\bar{g}$ can depart from the blockage station $i'$ on time, i.e., the actual departure time $d_{\bar{g},i'}$ equals $\bar{d}_{\bar{g},i'}$. As a result, train $\bar{g}$ will be delayed. Instead, the other trains must leave after $t_{\text{end}}$. Because train $g'$ first leaves from station $i'$, the actual departure time $d_{g',i'}$ equals $t_{\text{end}}$. Note

Table 1. Parameters of the model formulation.

| | Notation | Definition |
|---|---|---|
| Parameters | $G$ | total number of trains |
| | $I$ | total number of stations |
| | $g$ | train index, $g \in \{1, 2, ..., G\}$ |
| | $i$ | station index, $i \in \{1, 2, ..., I\}$ |
| | $i'$ | index of the blockage station, $i' \in \{1, 2, ..., I\}$ |
| | $t_{\text{start}}$ | start time of the station blockage |
| | $t_{\text{end}}$ | end time of the station blockage |
| | $Y$ | objective function |
| | $\bar{a}_{g,i}$ | planned arrival time for train $g$ at station $i$ |
| | $\bar{d}_{g,i}$ | planned departure time for train $g$ at station $i$ |
| | $g'$ | index of the first train departing from the blockage station $i'$, $g' \in \{1, 2, ..., G\}$ |
| | $\bar{g}$ | index of the train not affected by the station blockage, $\bar{g} \in M_{\text{no}}$ |
| | $g''$ | index of the train affected by the station blockage, $g'' \in M_{\text{blk}}$ |
| | $\bar{d}_{\bar{g},i'}$ | planned departure time for train $\bar{g}$ at station $i'$ |
| | $d_{\bar{g},i'}$ | actual departure time for train $\bar{g}$ at station $i'$ |
| | $\bar{d}_{g',i'}$ | planned departure time for train $g'$ at station $i'$ |
| | $d_{g',i'}$ | actual departure time for train $g'$ at station $i'$ |
| | $\bar{d}_{g'',i'}$ | planned departure time for train $g''$ at station $i'$ |
| | $d_{g'',i'}$ | actual departure time for train $g''$ at station $i'$ |
| | $h^{\text{dd}}_{g''-1,g'',i'}$ | depart-depart headway for train $g''-1$ and $g''$ at station $i'$ |
| | $s_{g,i}$ | actual dwell time for train $g$ at station $i$ |
| | $s^{\min}_{g,i}$ | minimum dwell time for train $g$ at station $i$ |
| | $h^{\text{aa}}_{g,g+1,i}$ | actual arrive-arrive headway for train $g$ and $g+1$ at station $i$ |
| | $h^{\text{dd}}_{g,g+1,i}$ | actual depart-depart headway for train $g$ and $g+1$ at station $i$ |
| | $h^{\text{ad}}_{g,g+1,i}$ | actual arrive-depart headway for train $g$ and $g+1$ at station $i$ |
| | $h^{\text{aa,min}}_{g,g+1,i}$ | minimum arrive-arrive headway for train $g$ and $g+1$ at station $i$ |
| | $h^{\text{dd,min}}_{g,g+1,i}$ | minimum depart-depart headway for train $g$ and $g+1$ at station $i$ |
| | $h^{\text{ad,min}}_{g,g+1,i}$ | minimum arrive-depart headway for train $g$ and $g+1$ at station $i$ |
| | $r_{g,i}$ | actual running time for train $g$ from station $i$ to $i+1$ |
| | $r^{\min}_{g,i}$ | minimum running time for train $g$ from station $i$ to $i+1$ |
| Decision variables | $a_{g,i}$ | actual arrival time for train $g$ at station $i$ |
| | $d_{g,i}$ | actual departure time for train $g$ at station $i$ |
| Sets | $M_{\text{no}}$ | set of trains not affected by the station blockage, $M_{\text{no}} = \{1, 2, ..., g'-1\}$ |
| | $M_{\text{blk}}$ | set of trains affected by the station blockage, $M_{\text{blk}} = \{g'+1, g'+2, ..., G\}$ |

that train $g''$ ($g'' \in M_{\text{blk}}$) stands at station $i'$, where $M_{\text{blk}} = \{g'+1, g'+2, ..., G\}$ is the set of the trains affected by the blockage except for train $g'$. The actual departure time $d_{g'',i'}$ equals the planned departure time $\bar{d}_{g'',i'}$ plus the depart-depart headway $h^{\text{dd}}_{g''-1,g'',i'}$, which is to guarantee a safe operation between the two successive trains $g''-1$ and $g''$ at station $i'$.

#### 2.2.2. *Constraints of train operation*

During the actual train rescheduling process, the regular operation that all trains should comply with usually consists of arrival, departure, stop, or pass at a station and run, tracking in a block section. Firstly, the constraint of the actual dwell time $s_{g,i}$ for train $g$ at station $i$ is presented by

$$s_{g,i} = d_{g,i} - a_{g,i} \geq s^{\min}_{g,i} \tag{3}$$

for $a_{g,i} \geq \bar{a}_{g,i}$ and $d_{g,i} \geq \bar{d}_{g,i}$ according to Assumption 1). The actual dwell time $s_{g,i}$ should satisfy the constraint of the minimum value $s_{g,i}^{\min}$ to ensure the operation of alighting and boarding passengers.

To guarantee the two successive trains $g$ and $g+1$ to enter or depart from the current station $i$ safely, the three types of headways (arrive-arrive headway $h_{g,g+1,i}^{\text{aa}}$, depart-depart headway $h_{g,g+1,i}^{\text{dd}}$, arrive-depart headway $h_{g,g+1,i}^{\text{ad}}$) should all satisfy the minimum value, which can be formulated as

$$h_{g,g+1,i}^{\text{aa}} = a_{g+1,i} - a_{g,i} \geq h_{g,g+1,i}^{\text{aa,min}} \tag{4}$$

$$h_{g,g+1,i}^{\text{dd}} = d_{g+1,i} - d_{g,i} \geq h_{g,g+1,i}^{\text{dd,min}} \tag{5}$$

$$h_{g,g+1,i}^{\text{ad}} = a_{g+1,i} - d_{g,i} \geq h_{g,g+1,i}^{\text{ad,min}} \tag{6}$$

for $g, g+1 \in \{1, 2, ..., G\}$ and $i \in \{1, 2, ..., I\}$, where $a_{g+1,i}$ and $d_{g+1,i}$ are the arrival time and departure time for train $g+1$ at station $i$, respectively. The variable $h_{g,g+1,i}^{\text{aa,min}}$, $h_{g,g+1,i}^{\text{dd,min}}$, and $h_{g,g+1,i}^{\text{ad,min}}$ are the minimum values of the corresponding headways.

The running time $r_{g,i}$ for train $g$ from station $i$ to $i+1$ also has the minimum value due to the constrain of train dynamics, which can be described as

$$r_{g,i} = a_{g,i+1} - d_{g,i} \geq r_{g,i}^{\min} \tag{7}$$

where $r_{g,i}^{\min}$ is the minimum running time and $a_{g,i+1}$ is the arrival time for train $g$ at station $i+1$.

*2.3. HSTR model*

To maintain a trade-off between model accuracy and algorithm computation speed, we establish a macroscopic integer linear programming (ILP) model for HSTR based on the train operation constraints (see Eqs. (2)–(7)). The HSTR model is formulated as follows:

$$\begin{aligned} &\min Y \\ &\text{s.t.} \quad (2) - (7) \\ &\qquad g, \bar{g}, g', g'', g''-1, g+1 \in \{1, 2, ..., G\} \\ &\qquad i, i', i+1 \in \{1, 2, ..., I\}. \end{aligned} \tag{8}$$

## 3. PS-SEGA for HSTR

HSTR is an NP-hard problem with train operation constraints [1, 2]. It is difficult to address the NP-hard problem merely using EAs, which may cause lower quality and efficiency for some solutions. However, EAs can acquire a quasi-optimal or even optimal solution in real time with appropriate improvement strategies. So we propose an efficient evolutionary algorithm called PS-SEGA for HSTR based on the ILP model (see Eq. (8) in Section 2.3). The proposed PS-SEGA considers the NP-hard feature of HSTR and combines train rescheduling characteristics with the advantage of EAs. Firstly, the basic parameters of EAs are prepared. The initial population will be generated by the hybrid initialization method. Then, the genetic process involving selection, crossover, and mutation will be carried out until the maximum generation number is reached. The heuristic decoding method will output arrival and departure times, and calculate the objective value for each generation under the partial station blockage. In addition, the restart strategy will be executed if the current population loses diversity. Finally, the optimal individual will be recorded. The corresponding rescheduled timetable will be output by the heuristic decoding method. Overall, the pseudocode and flow diagram of PS-SEGA are described in Algorithm 1 and Fig. 2. The main improvements of PS-SEGA are depicted as follows.

*3.1. Encoding method*

The encoding method is used to input decision variables of the HSTR model to PS-SEGA. Firstly, we introduce a traditional integer-based encoding method to represent the population individual of PS-SEGA. To improve the computation efficiency, we propose a novel permutation-based encoding method to define an efficient search space.

**Algorithm 1** PS-SEGA

Input: The planned arrival time $\bar{a}_{g,i}$ and planned departure time $\bar{d}_{g,i}$; The time period $[t_{\text{start}}, t_{\text{end}})$ under the partial station blockage;

Output: The actual arrival time $a_{g,i}$ and actual departure time $d_{g,i}$ after reordering trains at station $i'$;

Set the population size $N_{pop}$ in the encoding method;
Initialize the maximum number $NFE_{\max}$ of function evaluation, the restart threshold $\varepsilon$; Record the current number $m$ of function evaluation;
**for** $m$ = 1 to $NFE_{\max}$ **do**
    Initialize the initial population $pop$ with train departure sequences using Algorithm 3;
    Select parent individuals from $pop$; Perform crossover, mutation to produce offspring individuals; Merge the above parent and offspring individuals;
    Calculate the objective values (see Eq. (1) in Section 2.1) of the current individuals using the decoding method (see Algorithm 2);
    Calculate the fitness of the current individuals;
    Generate a new population $pop$ with the size of $N_{pop}$ by the fitness ranking method;
    **if** the diversity of the current population $pop$ is less than $\varepsilon$ **then**
        Perform the restart strategy by Algorithm 4;
    **end if**
**end for**
    Record the optimal individual and decode it using Algorithm 2 to output the train timetable (i.e., $a_{g,i}$ and $d_{g,i}$);
**return**

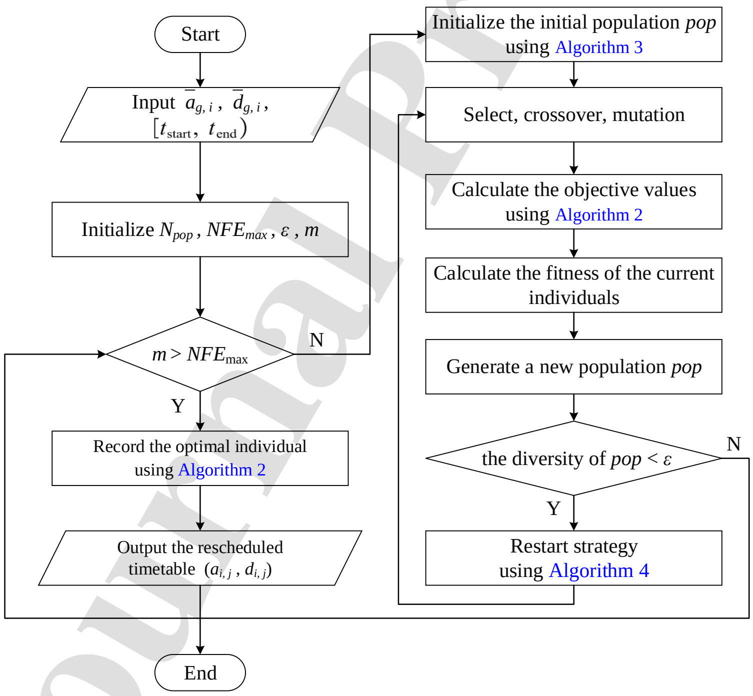



Figure 2. Flow diagram of PS-SEGA.

#### 3.1.1. Integer-based encoding

The essence of HSTR is to minimize the total train delay of trains by adjusting the arrival and departure times of each train at each station. In this way, the existing integer-based encoding method is the most straightforward method that uses arrival and departure times to denote individuals for EAs [17, 38]. So a population individual in the integer-based encoding method can be designed as

$$\mathbf{x}_{\text{IN}} = \left[a_{1,1}, d_{1,1}, \ldots, a_{g',i'}, d_{g',i'}, \ldots, a_{G,I}, d_{G,I}\right] \tag{9}$$

with $g' \in \{1, 2, \ldots, G\}$ is the first train departing from the blockage station $i'$ ($i' \in \{1, 2, \ldots, I\}$), where $\mathbf{x}_{\text{IN}}$ is a vector of arrival and departure times at all stations along the line. The dimension of $\mathbf{x}_{\text{IN}}$ is $2 \cdot G \cdot I$. Since station 1 and $G$ are the origin station and terminus, the arrival time at station 1 and the departure time at station $G$ do not exist. For similarity, let $a_{1,1} = d_{1,1}$, $d_{G,I} = a_{G,I}$. When station $i'$ blocks, the arrival and departure times $(a_{g',i'}, d_{g',i'}, \ldots, a_{G,I}, d_{G,I})$ after $t_{\text{start}}$ should move backward. PS-SEGA is used to adjust the train arrival and departure times in the representation of the integer-based encoding method.

#### 3.1.2. Permutation-based encoding

With the increase of trains and stations, the computation time of traditional integer-based encoding in solving HSTR increases exponentially. The existing method cannot guarantee the real-time performance of the rescheduling strategy. Therefore, combing the characteristic of HSTR and the advantage of evolutionary algorithms, we propose a novel permutation-based encoding method. According to Assumption 3), the rescheduling strategy of reordering trains only occurs at the blockage station $i'$. Hence, we can merely consider the train departure sequence at station $i'$. In this way, we propose a permutation-based encoding method by defining an efficient search space to address HSTR. The population individual in the permutation-based encoding method is defined as

$$\mathbf{x}_{\text{PE}} = \left[c_{1,i'}, c_{2,i'}, \ldots, c_{g,i'}, ..., c_{G-1,i'}, c_{G,i'}\right] \tag{10}$$

where $\mathbf{x}_{\text{PE}}$ is a vector denoting the train index $(c_{1,i'}, c_{2,i'}, \ldots, c_{g,i'}, \ldots, c_{G-1,i'}, c_{G,i'})$ at the blockage station $i'$. All the train indexes form a train departure sequence at the station $i'$. Therefore, the adjustment range of search space that PS-SEGA considers is reduced from 1440 minutes daily to the number of trains $G$. The novel encoding method can avoid searching infeasible and unnecessary search space. Finally, different individuals (i.e., different train departure sequences) form a new population, which will continue to be selected, crossed, and mutated by PS-SEGA until the maximum generation number is reached.

### 3.2. Decoding method

HSTR is essentially a constrained optimization problem [39]. The traditional decoding method with integer-based encoding can directly converted the population individuals into the train rescheduled timetable. However, this method requires handling all train operation constraints and consumes too much time. Therefore, we propose a heuristic decoding method to eliminate all the HSTR constraints (see Eqs. (2)–(7) in Section 2) to improve the computation efficiency. The train departure sequence obtained by the permutation-based encoding is denoted as the input to the decoding method. Significantly, the heuristic decoding method makes trains operate under the minimum value of train operation time, as depicted in Assumption 4). This train operation control mode is called “train tracking interval control” [40]. With the planned arrival and departure times in the departure sequence given by PS-SEGA, the process of the decoding method in eliminating constraints is described in the following steps.

1) Check the depart-depart headways of all influenced trains at the blockage station $i'$ and subsequent stations to find if constraints Eq. (2) and Eq. (5) are satisfied, respectively. If not, adjust arrival and departure times at station $i'$ and subsequent stations according to the minimum depart-depart headway $h^{\text{dd,min}}_{g,g+1,i'}$;

2) Check the arrive-arrive headways of all influenced trains whether to meet constraint Eq. (4). If not, adjust arrival and departure times under the minimum arrive-arrive headway $h^{\text{aa,min}}_{g,g+1,i'}$.

3) The arrival time $a_{g+1,i}$ of the latter train $g+1$ is determined by the arrival time $a_{g,i}$ of the former train $g$ and $h^{\text{aa,min}}_{g,g+1,i'}$. Therefore, the running time constraint Eq. (7) will be satisfied.

**Algorithm 2** Heuristic Decoding

Input: The planned arrival time $\bar{a}_{g,i}$ and planned departure time $\bar{d}_{g,i}$; The planned arrival time $\bar{a}^{\mathrm{PE}}_{g,i}$ and planned departure time $\bar{d}^{\mathrm{PE}}_{g,i}$ in the departure sequence calculated by PS-SEGA;

Output: The actual arrival time $a_{g,i}$ and actual departure time $d_{g,i}$ in the "train tracking interval control" mode;

1: **for** $g = 1$ to $G - 1$ **do**<br>
2:     **for** $i = i'$ to $I$ **do**<br>
3:         **if** $d_{g+1,i'} - d_{g,i'} < h^{\mathrm{dd,min}}_{g,g+1,i'}$ **then**<br>
4:             $d_{g+1,i} = d_{g,i} + h^{\mathrm{dd,min}}_{g,g+1,i}$;<br>
5:         **end if**<br>
6:     **end for**<br>
7: **end for**<br>
8: **for** $g = 1$ to $G$ **do**<br>
9:     **for** $i = i' + 1$ to $I$ **do**<br>
10:         $a_{g,i} = \bar{a}^{\mathrm{PE}}_{g,i} + \left(d_{g,i'} - \bar{d}^{\mathrm{PE}}_{g,i'}\right)$;<br>
11:         $d_{g,i} = \bar{d}^{\mathrm{PE}}_{g,i} + \left(d_{g,i'} - \bar{d}^{\mathrm{PE}}_{g,i'}\right)$;<br>
12:     **end for**<br>
13: **end for**<br>
14: **for** $g = 1$ to $G - 1$ **do**<br>
15:     **for** $i = i'$ to $I$ **do**<br>
16:         **if** $a_{g+1,i} - a_{g,i} < h^{\mathrm{aa,min}}_{g,g+1,i}$ **then**<br>
17:             $a_{g+1,i} = a_{g,i} + h^{\mathrm{aa,min}}_{g,g+1,i}$;<br>
18:         **end if**<br>
19:         **if** $d_{g+1,i} - d_{g,i} < h^{\mathrm{dd,min}}_{g,g+1,i}$ **then**<br>
20:             $\bar{s}_{g+1,i} = \bar{d}^{\mathrm{PE}}_{g+1,i} - \bar{a}^{\mathrm{PE}}_{g+1,i}$;<br>
21:             $d_{g+1,i} = \max\left\{d_{g,i} + h^{\mathrm{dd,min}}_{g,g+1,i}, a_{g+1,i} + \bar{s}_{g+1,i}\right\}$;<br>
22:         **end if**<br>
23:     **end for**<br>
24: **end for**<br>
25: **return**

4) The departure time $d_{g+1,i}$ for the latter train $g + 1$ is decided by the departure time $d_{g,i}$ of the former train $g$ and $h^{\mathrm{dd,min}}_{g,g+1,i'}$. Therefore, the dwell time constraint Eq. (3) will be satisfied.

Because trains run according to the minimum dwell times (see Assumption 4)), the arrive-depart headway constraint Eq. (6) is satisfied under the constraint of $h^{\mathrm{aa,min}}_{g,g+1,i'}$ and $h^{\mathrm{dd,min}}_{g,g+1,i'}$. Following the above constraint elimination operation, the pseudocode of the decoding method for rescheduling timetable is presented in Algorithm 2. Finally, the objective value and the rescheduled timetable will be output by the decoding method based on the departure sequence calculated by PS-SEGA.

**Remark 2:** In line17 of Algorithm 2, the arrival time $a_{g+1,i}$ (for train $g + 1$ at station $i$) is equal to the arrival time $a_{g,i}$ (for train $g$ at station $i$) and the minimum arrive-arrive headway $h^{\mathrm{aa,min}}_{g,g+1,i}$. Hence, using the interval buffer time to accelerate, train $g + 1$ may add a stop at station $i$ in the rescheduled timetable. Assume that train $g + 1$ will not completely stop at station $i$. Otherwise, train $g + 1$ will increase an additional start and stop time. In practice, train $g + 1$ can pass station $i$ at a lower speed.

**Remark 3:** When adjusting departure times, as shown from line 19 to 21 of Algorithm 2, we should add the planned dwell time $\bar{s}_{g+1,i}$ if train $g + 1$ has a stop at station $i$. That is because we cannot change the operation type for train $g + 1$ at station $i$ from the original stop to pass. If this happens, passengers cannot board at station $i$.

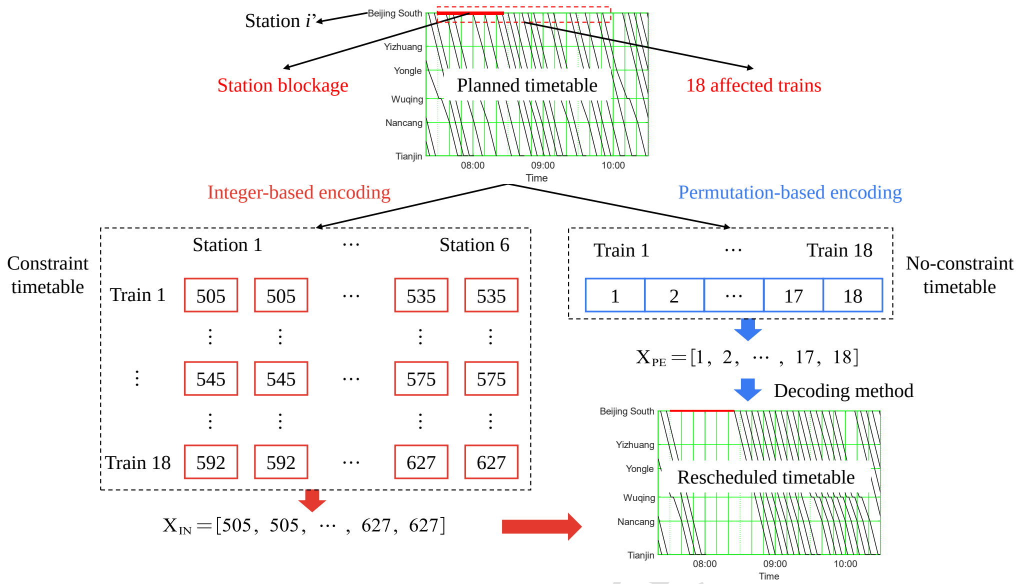



Figure 3. Comparison of the two encoding methods with the corresponding decoding method.

We present a numerical example to illustrate that the permutation-based encoding method can significantly reduce the solution space. Fig. 3 shows how the two encoding methods denote the train timetable. In the planned timetable with six stations, the 18 trains are affected by the partial station blockage from 7:30 to 8:25. With the permutation-based encoding and decoding method, we directly consider the train departure sequence at the blockage station and can obtain a no-constraint train timetable. The solution space under the two encoding methods is compared as follows. The maximum time horizon of rescheduling the timetable is up to 4 hours (i.e., 240 min), so dispatchers do not have to adjust all trains for a day at once. The solution space of the integer-based encoding method is the entire time domain (240 min) within the stage plan of 18 affected trains, i.e., $240^{2\times18\times6}$. However, the solution space of the permutation-based encoding method is the total number of all train departure sequences at the blockage station $i'$, i.e., 18!. Due to $18! < 18^{18} < 240^{18} < 240^{2\times18\times6}$, the solution space of the permutation-based encoding method is much smaller than that using the integer-based encoding method. This example verifies that the proposed permutation-based encoding method can reduce the solution space and improve the computation efficiency. Finally, the decoding method transforms the train departure sequence into the final rescheduled timetable.

### 3.3. Population initialization

The random initialization method provides an initial population with different individuals to keep good diversity. However, the random initialization method without problem-specific knowledge leads to low search efficiency. To accelerate the convergence speed of searching for a better strategy by PS-SEGA, we provide the initial population with a hybrid initialization method, including an efficient heuristic and the random initialization strategy.

Using problem-specific knowledge, the efficient heuristic strategy (EHS) generates an available departure sequence that produces a smaller total train delay at the blockage station. The problem-specific knowledge can be illustrated as follows: the shorter the running time of a train along the railway line, the higher priority for the train, which can depart earlier from the blockage station. Let $M_{\text{EHS}}$ represent the set of trains whose planned departure times are in the period $[t_{\text{start}}, t_{\text{end}})$. EHS only considers the influenced trains $M_{\text{EHS}}$ and generates the departure sequence $C_{\text{EHS}}$. The other influenced trains whose set is denoted as $M_{\text{FSFS}}$ are not in the time period $[t_{\text{start}}, t_{\text{end}})$. The variable $C_{\text{FSFS}}$ is denoted as the train departure sequence of the planned timetable for $M_{\text{FSFS}}$. Combining $C_{\text{EHS}}$ and $C_{\text{FSFS}}$, the complete departure sequence $C_{\text{ALL}}$ of all influenced trains is considered as an efficient population individual

**Algorithm 3** Hybrid Initialization

Input: The planned departure time $\bar{d}_{g',i'}$ and $\bar{d}_{g'',i'}$; The start and end time instants ($t_{\text{start}}$ and $t_{\text{end}}$) of the partial station blockage; The influenced trains $M_{\text{EHS}}$ in $[t_{\text{start}}, t_{\text{end}})$ and the other influenced trains $M_{\text{FSFS}}$;

Output: The initial population $pop$ after using EHS;

Let $M_{\text{EHS}}$ and $M_{\text{FSFS}}$ be the empty set;
**if** $t_{\text{start}} \le \bar{d}_{g'',i'} < t_{\text{end}}$ **then**
    Input train $g'$ and $g''$ to $M_{\text{EHS}}$;
    Generate the departure sequence $C_{\text{EHS}}$ of $M_{\text{EHS}}$ using EHS;
**else**
    Input train $g''$ to $M_{\text{FSFS}}$;
    Denote $C_{\text{FSFS}}$ of $M_{\text{FSFS}}$ as the departure sequence in the planned timetable;
**end if**
Combine $C_{\text{EHS}}$ and $C_{\text{FSFS}}$ to produce the complete departure sequence $C_{\text{ALL}}$ of all influenced trains;
Using $C_{\text{ALL}}$ to replace an individual of the random initial population;
**return**

**Algorithm 4** Restart Strategy

Input: The predefined threshold $\varepsilon$; The population size $N_{pop}$; The number of objective values $N_Y$;

Output: The new population $pop$ after using the restart strategy;

Record the optimal individual $pop'$ of the current population;
**if** $N_Y \big/ N_{pop} < \varepsilon$ **then**
    Initialize a new initial population called $pop''$ using random initialization with the size of $N_{pop} - 1$;
    Calculate the objective value and fitness of $pop''$;
    $pop = pop' + pop''$;
**end if**
**return**

to input to the initial population (see line 4 of Algorithm 1). Note that $g'$ is the first train leaving from the blockage station $i'$. The variable $g''$ denotes the train influenced by the station blockage, as shown in Eq. (2). Overall, the pseudocode of the hybrid initialization method using EHS is presented in Algorithm 3.

To illustrate the composition of the population individuals using the hybrid initialization method, we present a typical numerical example, as shown in Fig. 4. There are six influenced trains in the period $[t_{\text{start}}, t_{\text{end}})$ in the planned timetable. EHS generates the departure sequence of these six trains. According to the problem-knowledge of EHS, train 1, train 5, and train 6 have the same shortest running time along the line. So these three trains depart earlier from the blockage station $i'$. The running time of train 3 is shorter than that of the other three trains. Then, compare the value of the running and dwell time of the other two trains. We can know that train 4 departs earlier than train 2. So $C_{\text{EHS}}$ is (1, 5, 6, 3, 4, 2). The other 12 influenced trains not in the time period $[t_{\text{start}}, t_{\text{end}})$ follow the departure sequence of the planned timetable. $C_{\text{FSFS}}$ is equal to (7, 8, ..., 17, 18). Combing $C_{\text{EHS}}$ and $C_{\text{FSFS}}$, the complete departure sequence $C_{\text{ALL}}$ of all 18 influenced trains is (1, 5, 6, 3, 4, 2, 7, 8, ..., 17, 18).

### 3.4. *Restart strategy*

When PS-SEGA runs iteratively, the population converges to lower diversity. This means that most of the population individuals has the same objective value. The population may be trapped into a local optimum before finding the optimal individual.

To maintain population diversity, we propose a restart strategy. When the proportion of the current objective values in the population is less than the predefined threshold $\varepsilon$, we will rebuild this population with an optimal individual and random individuals, as shown in Algorithm 4. In this way, the species of population individuals recover diversity.

Fig. 5 is a numerical example of when and how to use the restart strategy to keep population diversity. Note that there are ten individuals in the current population. The circle in the same color indicates the individual with the

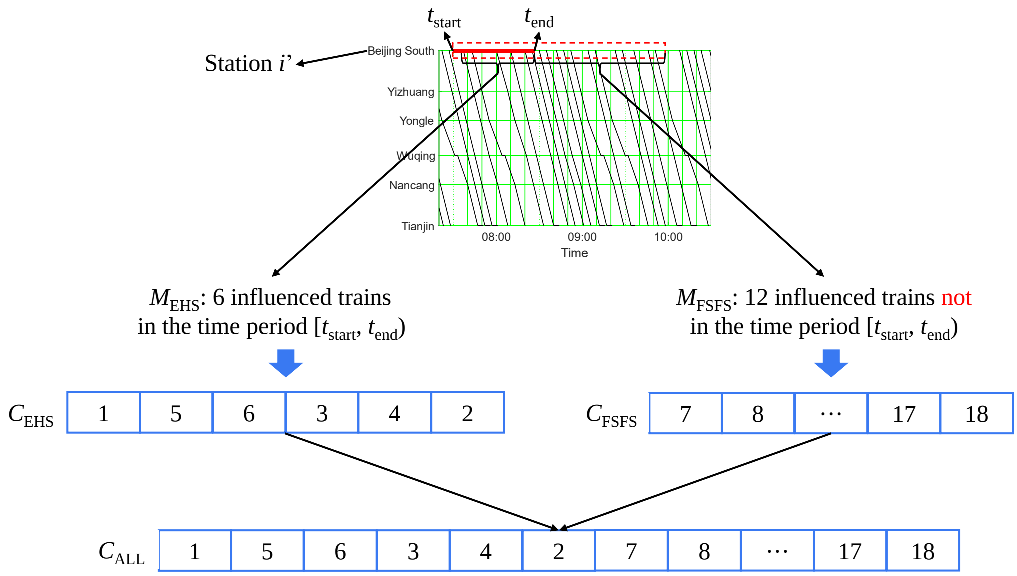



Figure 4. A typical numerical example of population initialization.

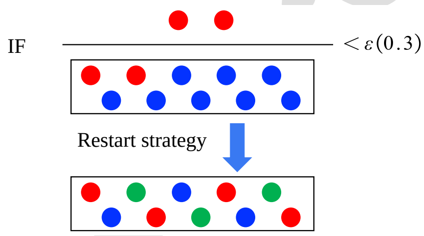



Figure 5. A typical example of restart strategy.

same objective value. So the proportion of the current objective values in the population is calculated as 0.2, which is smaller than $\varepsilon$ (0.3). As a result, the restart strategy should be executed to produce different individuals.

### *3.5. Illustrative example*

To better understand the PS-SEGA, we present a typical illustrative example, as shown in Fig. 6. We still consider 18 affected trains running at six stations along the HSTR line. As seen in Fig. 6, the initial population has an individual with a smaller total train delay using the hybrid initialization method. The random initialization method produces the other $N_{pop} - 1$ individuals. The genes (i.e., the train index at the blockage station $i'$) of all $N_{pop}$ population individuals are changed by selection, crossover, and mutation. The better offspring individuals are produced and selected when calculating the population fitness in the current function evaluation. If the proportion of the current objective value is less than $\varepsilon$, the restart strategy is performed to rebuild the current population. Finally, the decoding method transforms the optimal individual (i.e., the optimal train departure sequence at station $i'$) into the rescheduled timetable with the smallest total train delay.

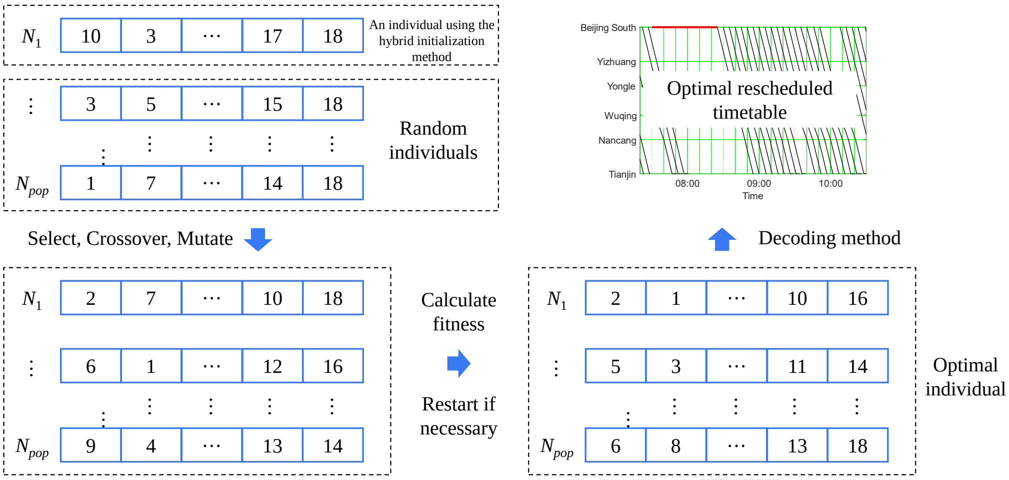



Figure 6. A typical illustrative example of the PS-SEGA.

Table 2. Parameters of the model formulation.

| Station Name | $l_i$/km | $r_{g,i}^{\min}$/min |
|---|---|---|
| Beijing South | 0 | 7 |
| Yizhuang | 22 | 5 |
| Yongle | 46 | 6 |
| Wuqing | 84 | 5 |
| Nancang | 108 | 7 |
| Tianjin | 120 | NA |

## 4. Numerical experiments

To illustrate the advantage and effectiveness of PS-SEGA's improvements, we set the partial station blockage in the train timetable to generate typical scenarios. A comparative study is performed between PS-SEGA and the other two efficient EAs and their improved variants also using PS-SEGA's improvements. Furthermore, intensive numerical experiments are carried out to demonstrate the superiority of the encoding/decoding and hybrid initialization methods in PS-SEGA. The experiments run in Python on a computer with an Intel Core i7-9700T CPU @2.00 GHz with 16 GB RAM.

### 4.1. Scenarios setup

In the numerical experiments, we consider the train timetable of the Beijing-Tianjin high-speed railway. The intermediate stations along the railway line are Yizhuang, Yongle, Wuqing, and Nancang. Notably, Nancang is a special railway station that does not handle passenger service and only divides block sections. Table 2 lists the position $l_i$ of all stations and the minimum running time $r_{g,i}^{\min}$ in all block sections of the Beijing-Tianjin high-speed railway. The minimum dwell time $s_{g,i}^{\min}$ is 2 min. The value of $h_{g,g+1,i}^{\mathrm{aa},\min}$, $h_{g,g+1,i}^{\mathrm{dd},\min}$, and $h_{g,g+1,i}^{\mathrm{ad},\min}$ are set to 5 min, 5 min, and 3 min, respectively.

As shown in Table 3, ten typical scenarios are installed at Beijing South with different station blockage durations. The reason for selecting this station is that the affected trains can be accommodated at Beijing South and the nearby rolling stock depot. Depending on whether the blockage duration exceeds 30 min, the ten scenarios can be divided into two quintessential types, i.e., five small disturbances No. 1–5 and five large disturbances No. 6–10. The other

Table 3. Scenarios under the partial station blockage at Beijing South.

| No. | $[t_{\text{start}}, t_{\text{end}})$ | $t_{\text{blk}}$/min | $N_{\text{blk}}$ | $N_{\text{con}}$ |
|---|---|---|---|---|
| 1 | [6:00, 6:20) | 20 | 4 | 5 |
| 2 | [7:10, 7:21) | 11 | 1 | 5 |
| 3 | [8:40, 9:10) | 30 | 5 | 9 |
| 4 | [11:50, 12:20) | 30 | 3 | 6 |
| 5 | [13:30, 14:00) | 30 | 3 | 8 |
| 6 | [7:30, 8:25) | 55 | 6 | 20 |
| 7 | [10:20, 11:00) | 40 | 5 | 13 |
| 8 | [12:50, 14:10) | 80 | 9 | 24 |
| 9 | [16:30, 18:30) | 120 | 14 | 36 |
| 10 | [17:30, 19:10) | 100 | 13 | 31 |

parameters in Table 3 include the time period $[t_{\text{start}}, t_{\text{end}})$ under the partial station blockage, the blockage duration $t_{\text{blk}}$, the number of affected trains ($N_{\text{blk}}$) in $[t_{\text{start}}, t_{\text{end}})$, and the number of considered trains ($N_{\text{con}}$) inputting to EAs.

### *4.2. Algorithms under comparison*

Classical evolutionary algorithm frameworks in existing encoding, decoding, or population initialization methods hardly provide rescheduling solutions in real time for the HSTR problem. Consequently, we propose an efficient evolutionary algorithm called PS-SEGA aiming at the characteristics of HSTR. To verify the advantage and applicability of PS-SEGA for solving HSTR, we compare PS-SEGA and its competitors, i.e.,

**SEGA** [41]: Genetic algorithm (GA) is a renowned and well-regarded algorithm to address scheduling or optimization problems. Some operations of choosing elite individuals can be performed for GA to improve its efficiency [42]. For example, strengthen elitist genetic algorithm (SEGA) is one of the most effective GA variants. Firstly, SEGA merges the parent individuals with the cross-variant individuals. Then, it selects the elite individuals from the merged individuals [41]. This operation can increase the possibility of selecting the optimal elite individual in each iteration.

**Differential Evolution (DE)** [43]: Many researchers have established the HSTR model based on the same NP-hard feature in the JSP problem [11, 44]. DE is a prevailing and efficient evolutionary algorithm to address the permutation-based JSP problem. Moreover, DE performs better than particle swarm optimization and its variants [45–47]. Therefore, DE is selected as the first compared evolutionary algorithm.

**Evolution Strategy (ES)** [48]: This is another efficient evolutionary algorithm that directly operates on the phenotype of the population. ES does not require modifying the gene. We choose $(\mu + \lambda)$-ES to resolve HSTR [48]. The calculation steps are as follows. Firstly, let $\mu$ parents of the current population produce $\lambda$ individuals. Then, select $(\mu + \lambda)$ individuals to participate in the competition. Finally, preserve $\mu$ optimal individuals.

The fundamental code of the three basic EAs (i.e., DE, SEGA, and ES) mentioned above has been provided by Jazzbin *et al.* in a toolbox called Geatpy [49], which was implemented with Python programming language. Based on Geatpy, we perform different combinations of population initialization methods and restart strategies on the three basic EAs to generate different improved EAs. To verify the optimality and efficiency of the improved EAs, we compare the rescheduling solution calculated by the improved EAs and CPLEX in each scenario, respectively.

In addition, the two heuristics algorithms, First-Scheduled-First-Served (FSFS) [15] and the proposed EHS (see Section 3.3), are presented to analyze the noticeable effect of reducing the total train delay using the improved EAs. Notably, the departure sequence calculated by FSFS is the same as that of the planned timetable.

**Remark 4:** Since SEGA and ES in Geatpy do not have the permutation-based encoding method, we employ an ingenious method called *random key* to transform the continuous space to the permutation space [50]. In this way, SEGA and ES can be used in the permutation-based encoding method successfully.

To compare the performance of the improved EAs aforementioned, we perform 20 independent experiments in each scenario in Table 3. Beforehand, the most appropriate parameters of the three basic EAs and the two encoding

Table 4. Parameters of two encoding methods and three evolutionary algorithms.

| Parameter | Value |
|---|---|
| Population size $N_{pop}$ | 50 |
| Maximum number of function evaluation $NFE_{\max}$ in the integer-based encoding | 10000 |
| Maximum number of function evaluation $NFE_{\max}$ in the permutation-based encoding | 400 |
| Predefined threshold $\varepsilon$ for the restart strategy | 0.02 |
| Crossover rate $p_c$ for DE | 0.5 |
| Crossover rate $p_c$ for SEGA | 0.7 |
| Mutation rate $p_m$ for DE | 0.5 |
| Mutation rate $p_m$ for SEGA | 0.5 |

Table 5. Abbreviations of encoding methods, population initialization methods and restart strategies.

| Abbreviations | Methods/Strategies |
|---|---|
| IN | Integer-based encoding method |
| PE | Permutation-based encoding method |
| RI | Random initialization method |
| HI | Hybrid initialization method |
| NR | No restart strategy |
| HR | Hybrid restart strategy |

methods require to be determined by sensitivity analysis. Therefore, we perform experiments under different parameter settings, i.e., $N_{pop}$ with 40, 45, 50, 55, 60, $NFE_{\max}$ in the integer-based encoding with 8000, 10000, 12000, $NFE_{\max}$ in the permutation-based encoding with 200, 400, 600, $\varepsilon$ with 0.02, 0.04, 0.06, $p_c$ and $p_m$ with 0.5, 0.6, 0.7, 0.8, 0.9. Table 4 lists the most appropriate parameters that can provide the best overall performance.

### 4.3. Results analysis

This section compares the two encoding methods and different combinations of population initialization methods with restart strategies. To illustrate that PS-SEGA can provide better solutions than its competitors in solving HSTR, we employ the Wilcoxon rank-sum test with a significance level of 5%. If one algorithm is superior to another algorithm on the objective value in comparison, $Score$ equals 1. Otherwise, $Score$ is 0.

For convenience, the abbreviations of encoding methods, population initialization methods, and restart strategies are indicated in Table 5. Notably, since the population of the restart strategy involves an optimal individual and random individuals, we also denote the restart strategy as the hybrid restart strategy (HR). Then, we can add suffixes (i.e., the abbreviations in Table 5) to the three basic EAs (i.e., DE, SEGA, and ES) to represent different improved EAs. For instance, PE-DE-RI-HR indicates differential evolution (DE) in the permutation-based (PE) encoding with the random initialization method (RI) and the hybrid restart strategy (HR). With the hybrid initialization method (HI) and the hybrid restart strategy (HR), the proposed PS-SEGA can also be denoted as PE-SEGA-HI-HR.

#### 4.3.1. Comparison of different encoding approaches

To illustrate the effectiveness of the permutation-based encoding method in PS-SEGA, we only employ the random initialization method (RI) without the restart strategy (NR) on SEGA in the permutation-based and integer-based encoding, respectively. In other words, we compare the performance of PE-SEGA-RI-NR and IN-SEGA-RI-NR.

Table 6. Abbreviations of encoding methods, population initialization methods and restart strategies.

| No. | PE-SEGA-RI-NR | | IN-SEGA-RI-NR | |
|---|---|---|---|---|
| | val/min | time/s | val/min | time/s |
| 1 | **1080.0 ± 0.0** | 2.06 ± 0.05 | NA | 402.06 ± 2.94 |
| 2 | **304.9 ± 8.1** | 2.27 ± 0.13 | NA | 402.32 ± 2.75 |
| 3 | **2036.0 ± 0.0** | 3.81 ± 0.06 | NA | 500.87 ± 3.41 |
| 4 | **1096.0 ± 0.0** | 2.72 ± 0.12 | NA | 427.593 ± 3.57 |
| 5 | **1532.0 ± 0.0** | 3.56 ± 0.08 | NA | 471.64 ± 5.42 |
| 6 | **4754.9 ± 44.7** | 8.37 ± 0.15 | NA | 776.04 ± 29.22 |
| 7 | **2407.9 ± 8.3** | 5.00 ± 0.13 | NA | 598.26 ± 4.39 |
| 8 | **10323.8 ± 83.3** | 10.72 ± 0.40 | NA | 869.82 ± 11.85 |
| 9 | **22768.4 ± 229.6** | 15.582 ± 0.586 | NA | 1132.28 ± 14.34 |
| 10 | **18805.4 ± 84.1** | 12.00 ± 0.49 | NA | 1009.34 ± 7.87 |

NA: There is no feasible solutions in the current algorithm.

Using the above two algorithms, we perform experiments in the ten scenarios (Table 3). The objective value and computation time (in terms of the mean and standard deviation) are listed in Table 6. The best results are all marked in bold.

As for the objective value and computation time (Table 6), PE-SEGA-RI-NR in the permutation-based encoding significantly outperforms IN-SEGA-RI-NR in the integer-based encoding in all scenarios. PE-SEGA-RI-NR can give the HSTR solutions with 0 standard deviations in the four small disturbance scenarios (i.e., No. 1, No. 3, No. 4, and No. 5). However, this does not mean that the above four scenarios are too simple because IN-SEGA-RI-NR cannot address them at all. This result illustrates the great advantage of the permutation-based encoding method in PS-SEGA because of the defined efficient search space based on the objective of HSTR. To further validate this advantage on computation time, we can see from Table 6 that IN-SEGA-RI-NR cannot address all ten scenarios in enough time. This means that the integer-based encoding method's search efficiency and solution quality are not ideal. It is difficult to obtain feasible solutions with enough function evaluation. However, PE-SEGA-RI-NR can address the same ten scenarios within 20 s. Overall, the permutation-based encoding method can efficiently address HSTR by defining an efficient search space based on the feature and objective of the HSTR problem.

*4.3.2. Comparison of different combinations of population initialization methods and restart strategies*

To verify the performance of the improvements in PS-SEGA, we perform different combinations of population initialization methods and restart strategies 20 times on the three basic EAs (i.e., DE, SEGA, and ES). In terms of the mean and standard deviation, the optimized objective value (val) and the computation time (time) of each improved evolutionary algorithm are all reported in Tables 7–9. The best results are all marked in bold. The parameter $Score$ is used to measure the performance of each improved evolutionary algorithm. Because 120 comparisons will be made in the Wilcoxon rank-sum test for each improved evolutionary algorithm, the maximum value of $Score$ is 120. Since all the improved EAs can provide the HSTR solution close to the optimal solution, there is not much difference in $Score$ of some improved EAs.

(1) As for DE and its improved variants (Table 7), PE-DE-HI-NR only performing the hybrid initialization method obtains the highest $Score$. This means the hybrid initialization method can improve the performance of DE. Instead, the restart strategy (PE-DE-RI-HR's $Score$ is 21) reduces the performance of the no-improved DE (PE-DE-NI-NR's $Score$ is 24). This result means DE has a robust global search capability and does not necessarily use the restart strategy.

(2) For SEGA and its improved variants (Table 8), the three improved SEGA (PE-SEGA-HI-NR, PE-SEGA-RI-HR, and PS-SEGA) can eliminate the standard deviation of the objective values in the scenarios of No. 2 and No. 7. This means that the hybrid initialization method and the restart strategy can enhance SEGA's performance increasingly. Though the two improved EAs, PE-SEGA-HI-NR and PS-SEGA, have the same value of $Score$, PS-SEGA, using the restart strategy, obtains the minimum objective values in all ten scenarios. We can conclude that the

Table 7. Comparison results on different combinations of population initialization methods and restart strategies on DE.

| No. | PE-DE-RI-NR | | PE-DE-HI-NR | | PE-DE-RI-HR | | PE-DE-HI-HR | |
|---|---|---|---|---|---|---|---|---|
| | val/min | time/s | val/min | time/s | val/min | time/s | val/min | time/s |
| 1 | **1080.0 ± 0.0** | 2.04 ± 0.07 | **1080.0 ± 0.0** | 1.94 ± 0.07 | **1080.0 ± 0.0** | 2.16 ± 0.07 | **1080.0 ± 0.0** | 2.15 ± 0.11 |
| 2 | **303.0 ± 0.0** | 2.23 ± 0.12 | **303.0 ± 0.0** | 2.06 ± 0.07 | **303.0 ± 0.0** | 2.20 ± 0.16 | **303.0 ± 0.0** | 2.06 ± 0.07 |
| 3 | **2036.0 ± 0.0** | 3.83 ± 0.07 | **2036.0 ± 0.0** | 3.63 ± 0.07 | **2036.0 ± 0.0** | 3.75 ± 0.15 | **2036.0 ± 0.0** | 3.78 ± 0.03 |
| 4 | **1096.0 ± 0.0** | 2.75 ± 0.11 | **1096.0 ± 0.0** | 2.64 ± 0.05 | **1096.0 ± 0.0** | 2.73 ± 0.06 | **1096.0 ± 0.0** | 2.71 ± 0.05 |
| 5 | **1532.0 ± 0.0** | 3.56 ± 0.09 | **1532.0 ± 0.0** | 3.49 ± 0.06 | **1532.0 ± 0.0** | 3.56 ± 0.04 | **1532.0 ± 0.0** | 3.53 ± 0.06 |
| 6 | 4725.6 ± 1.2 | 8.40 ± 0.12 | **4725.0 ± 0.0** | 8.30 ± 0.12 | 4726.2 ± 2.0 | 8.48 ± 0.09 | **4725.0 ± 0.0** | 8.24 ± 0.16 |
| 7 | **2406.0 ± 0.0** | 4.95 ± 0.14 | **2406.0 ± 0.0** | 5.10 ± 0.11 | **2406.0 ± 0.0** | 5.13 ± 0.08 | **2406.0 ± 0.0** | 5.12 ± 0.07 |
| 8 | 10208.2 ± 18.6 | 10.83 ± 0.39 | 10205.7 ± 16.1 | 10.87 ± 0.33 | 10221.8 ± 24.6 | 11.00 ± 0.35 | 10207.1 ± 16.8 | 10.78 ± 0.25 |
| 9 | 22657.1 ± 14.8 | 16.07 ± 0.55 | 22638.7 ± 7.7 | 15.88 ± 0.51 | 22669.2 ± 23.5 | 16.05 ± 0.66 | 22635.9 ± 10.7 | 15.73 ± 0.54 |
| 10 | 18682.7 ± 25.9 | 12.38 ± 0.45 | 18660.5 ± 21.3 | 11.94 ± 0.28 | 18676.1 ± 25.6 | 12.43 ± 0.34 | 18661.9 ± 20.0 | 12.31 ± 0.30 |
| *Score* | 24 | | 31 | | 21 | | 30 | |

Table 8. Comparison results on different combinations of population initialization methods and restart strategies on SEGA.

| No. | PE-SEGA-RI-NR | | PE-SEGA-HI-NR | | PE-SEGA-RI-HR | | PS-SEGA | |
|---|---|---|---|---|---|---|---|---|
| | val/min | time/s | val/min | time/s | val/min | time/s | val/min | time/s |
| 1 | **1080.0 ± 0.0** | 2.06 ± 0.05 | **1080.0 ± 0.0** | 2.02 ± 0.07 | **1080.0 ± 0.0** | 2.58 ± 0.16 | **1080.0 ± 0.0** | 2.55 ± 0.13 |
| 2 | 304.9 ± 8.1 | 2.27 ± 0.13 | **303.0 ± 0.0** | 2.08 ± 0.06 | **303.0 ± 0.0** | 2.51 ± 0.14 | **303.0 ± 0.0** | 2.41 ± 0.29 |
| 3 | **2036.0 ± 0.0** | 3.81 ± 0.06 | **2036.0 ± 0.0** | 3.63 ± 0.07 | **2036.0 ± 0.0** | 4.08 ± 0.08 | **2036.0 ± 0.0** | 4.11 ± 0.17 |
| 4 | **1096.0 ± 0.0** | 2.72 ± 0.12 | **1096.0 ± 0.0** | 2.62 ± 0.04 | **1096.0 ± 0.0** | 2.87 ± 0.04 | **1096.0 ± 0.0** | 2.88 ± 0.05 |
| 5 | **1532.0 ± 0.0** | 3.56 ± 0.08 | **1532.0 ± 0.0** | 3.52 ± 0.05 | **1532.0 ± 0.0** | 4.01 ± 0.04 | **1532.0 ± 0.0** | 3.98 ± 0.03 |
| 6 | 4754.9 ± 44.7 | 8.37 ± 0.15 | 4727.7 ± 0.9 | 8.30 ± 0.13 | **4727.0 ± 1.4** | 9.17 ± 0.08 | **4727.0 ± 1.4** | 9.25 ± 0.14 |
| 7 | 2407.9 ± 8.3 | 5.00 ± 0.13 | **2406.0 ± 0.0** | 5.15 ± 0.11 | **2406.0 ± 0.0** | 5.44 ± 0.22 | **2406.0 ± 0.0** | 5.53 ± 0.16 |
| 8 | 10323.8 ± 83.3 | 10.72 ± 0.40 | 10199.2 ± 30.6 | 10.33 ± 0.34 | 10224.5 ± 39.1 | 11.68 ± 0.39 | **10188.9 ± 11.8** | 11.57 ± 0.26 |
| 9 | 22768.4 ± 229.6 | 15.58 ± 0.59 | 22620.7 ± 28.0 | 15.23 ± 0.47 | 22760.6 ± 148.1 | 16.64 ± 0.92 | **22620.5 ± 19.4** | 16.92 ± 0.63 |
| 10 | 18805.4 ± 84.1 | 12.00 ± 0.49 | 18660.1 ± 40.0 | 11.27 ± 0.16 | 18727.0 ± 60.8 | 13.22 ± 0.53 | **18647.4 ± 25.7** | 12.98 ± 0.28 |
| *Score* | 4 | | **35** | | 19 | | **35** | |

restart strategy can improve SEGA's performance. The most efficient improved variant of SEGA is PS-SEGA.

(3) As for ES and its improved variants (Table 9), PE-ES-HI-HR is superior to the other improved ES but not so much. The hybrid population initialization method and the restart strategy cannot improve ES significantly, which means the improvements of PS-SEGA are not all suitable for ES. Overall, the most efficient improved variants of DE, SEGA, and ES are PE-DE-HI-NR, PS-SEGA, and PE-ES-HI-HR, respectively.

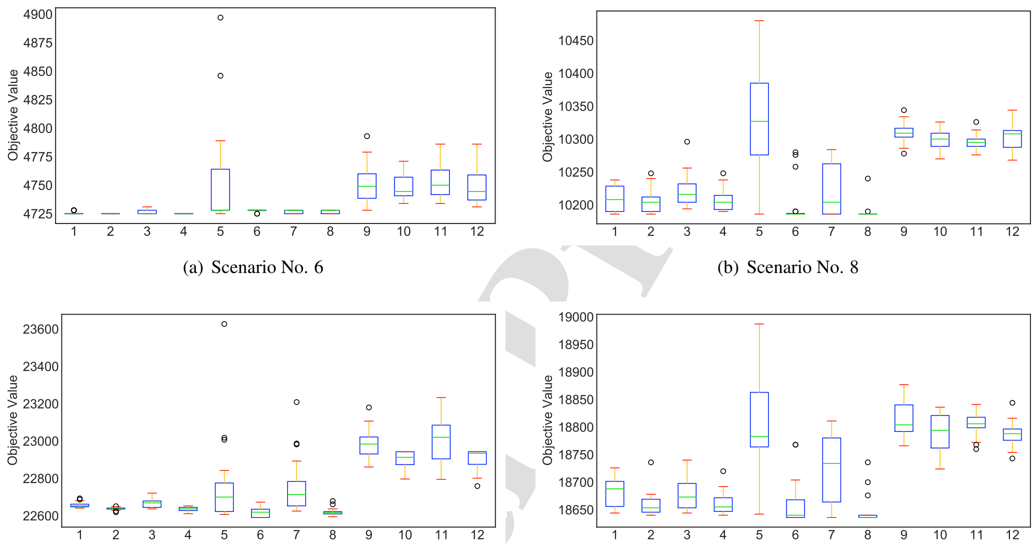
All the improved EAs can provide optimized solutions in all of the ten scenarios in less than 1 min (Tables 7–9), which meets the real-time requirement of HSTR. ES and its improved variants perform the worst but has the least computation time in resolving HTRS. DE, SEGA, and their improved variants have a longer computation time, but all perform better. Besides, the objective values for each improved evolutionary algorithm have significant differences in the scenarios of No. 6, No. 8, No. 9, and No. 10. Hence, we select these four scenarios to draw the box plots, as displayed in Figs. 7(a)–7(d). The data distribution of the objective value for all improved EAs can be seen in Fig. 7. The circle represents outliers of the objective values. The stability of the improved EAs can be known from the box plots. For example, although there are also a few outliers in PE-SEGA under the four typical scenarios, the distribution of the objective values in PS-SEGA is relatively concentrated compared with the other improved EAs. We can conclude that PS-SEGA has the best stability.

#### *4.3.3. Convergence analysis*

Similarly, we also choose the four typical scenarios (No. 6, No. 8, No. 9, and No. 10) to draw the average objective value of 20 independent experiments for each generation in all improved EAs, as shown in Figs. 8(a)–8(d). The corresponding objective values in terms of the mean and standard deviation can be referred to in Tables 7–9.

Table 9. Comparison results on different combinations of population initialization methods and restart strategies on ES.

| No. | PE-ES–RI-NR val/min | PE-ES–RI-NR time/s | PE-ES–HI-NR val/min | PE-ES–HI-NR time/s | PE-ES–RI-HR val/min | PE-ES–RI-HR time/s | PE-ES–HI-HR val/min | PE-ES–HI-HR time/s |
|---|---|---|---|---|---|---|---|---|
| 1 | **1080.0 ± 0.0** | 1.42 ± 0.04 | **1080.0 ± 0.0** | 1.40 ± 0.05 | **1080.0 ± 0.0** | 1.77 ± 0.06 | **1080.0 ± 0.0** | 1.78 ± 0.11 |
| 2 | **303.0 ± 0.0** | 1.55 ± 0.11 | **303.0 ± 0.0** | 1.44 ± 0.05 | **303.0 ± 0.0** | 1.67 ± 0.11 | **303.0 ± 0.0** | 1.58 ± 0.08 |
| 3 | **2036.0 ± 0.0** | 2.39 ± 0.05 | **2036.0 ± 0.0** | 2.25 ± 0.06 | **2036.0 ± 0.0** | 2.40 ± 0.15 | **2036.0 ± 0.0** | 2.39 ± 0.28 |
| 4 | **1096.0 ± 0.0** | 1.83 ± 0.03 | **1096.0 ± 0.0** | 1.76 ± 0.045 | **1096.0 ± 0.0** | 1.888 ± 0.05 | **1096.0 ± 0.0** | 1.93 ± 0.07 |
| 5 | **1532.0 ± 0.0** | 2.30 ± 0.08 | **1532.0 ± 0.0** | 2.19 ± 0.05 | **1532.0 ± 0.0** | 2.36 ± 0.05 | **1532.0 ± 0.0** | 2.38 ± 0.05 |
| 6 | 4752.3 ± 16.7 | 4.89 ± 0.09 | **4748.6 ± 11.3** | 4.83 ± 0.08 | 4754.3 ± 14.7 | 4.78 ± 0.06 | 4749.3 ± 15.8 | 4.79 ± 0.06 |
| 7 | 2406.0 ± 0.0 | 3.04 ± 0.06 | **2406.0 ± 0.0** | 3.05 ± 0.06 | **2406.0 ± 0.0** | 3.03 ± 0.04 | **2406.0 ± 0.0** | 2.99 ± 0.05 |
| 8 | 10310.5 ± 15.9 | 5.95 ± 0.31 | 10298.9 ± 14.8 | 5.89 ± 0.23 | 10295.9 ± 11.1 | 6.13 ± 0.28 | 10303.8 ± 20.4 | 5.91 ± 0.25 |
| 9 | 22985.4 ± 76.7 | 8.76 ± 0.19 | 22900.9 ± 46.7 | 8.70 ± 0.21 | 23004.4 ± 116.5 | 8.76 ± 0.23 | 22902.1 ± 54.5 | 8.70 ± 0.18 |
| 10 | 18811.2 ± 29.4 | 7.11 ± 0.27 | 18789.6 ± 34.6 | 7.06 ± 0.27 | 18804.9 ± 21.9 | 7.17 ± 0.30 | 18787.2 ± 21.9 | 6.97 ± 0.25 |
| *Score* | 0 | | 3 | | 1 | | 4 | |




(a) Scenario No. 6

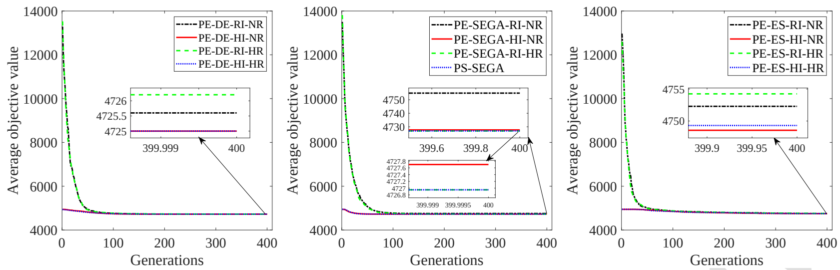
(b) Scenario No. 8

(c) Scenario No. 9

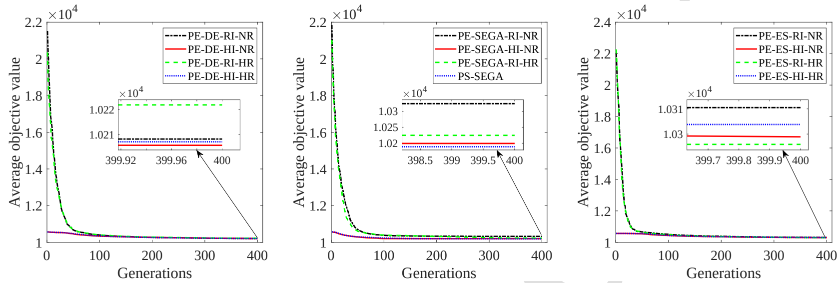
(d) Scenario No. 10

Figure 7. Box plots for different combinations of population initialization methods and restart strategies on DE, SEGA and ES under four typical scenarios. All improved EAs in Tables 7–9 are numbered from 1 to 12, i.e., 1: PE-DE-RI-NR, 2: PE-DE-HI-NR, 3: PE-DE-RI-HR, 4: PE-DE-HI-HR, 5: PE-SEGA-RI-NR, 6: PE-SEGA-HI-NR, 7: PE-SEGA-RI-HR, **8: PS-SEGA**, 9: PE-ES-RI-NR, 10: PE-ES-HI-NR, 11: PE-ES-RI-HR, 12: PE-ES-HI-HR.

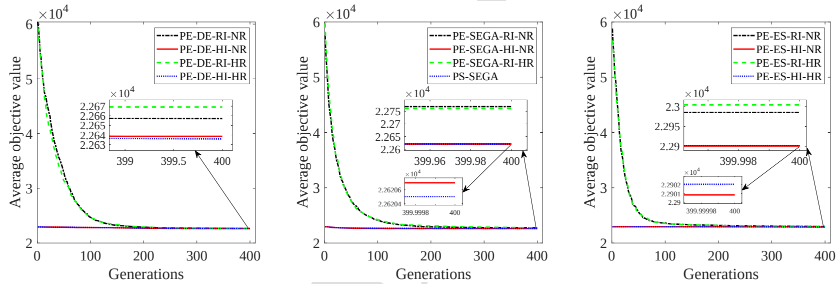
We can further confirm that the optimal improved variants of DE, SEGA, and ES are PE-DE-HI-NR, PS-SEGA, and PE-ES-HI-HR, respectively. In addition, all the improved EAs using the hybrid initialization (i.e., the solid red line and dotted blue line) can accelerate the convergence speed in the early stage of EAs and are faster to acquire a better solution. We can also clarify which optimal improved variant has the fastest convergence speed. For example, PS-SEGA (i.e., the dotted blue line) performing the hybrid initialization method and restart strategy converges faster than the other improved variants of SEGA in all four typical scenarios.



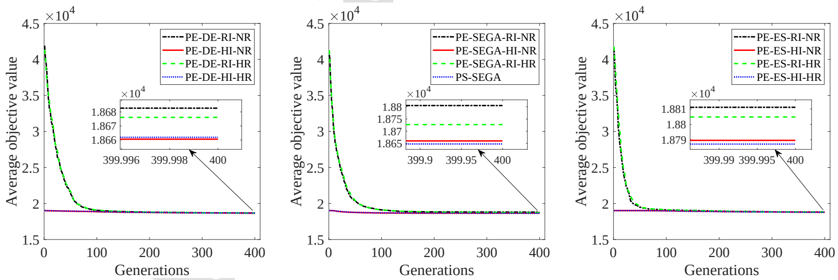

(a) Scenario No. 6




(b) Scenario No. 8




(c) Scenario No. 9




(d) Scenario No. 10

Figure 8. Convergence curves of average objective values on all improved EAs under four typical scenarios.

*4.3.4. Most efficient evolutionary algorithm, PS-SEGA*

According to the above analysis, we choose the optimal combination of population initialization methods and restart strategies on DE, SEGA, and ES to produce the three efficient EAs (i.e., PE-DE-HI-NR, PS-SEGA, and PE-ES-HI-HR). Then, we validate that PS-SEGA is the most efficient evolutionary algorithm from the three following indicators, as shown in Fig. 9.

(1) *Indicator 1: the degree of performance improvement.* Compared with the three no-improved EAs (PE-DE-RI-NR, PE-SEGA-RI-NR, PE-ES-RI-NR), the degrees of performance improvement on PE-DE-HI-NR, PS-SEGA, and PE-ES-HI-HR under the PS-SEGA improvements are 7, 31 and 4, respectively. Therefore, PE-SEGA has the best performance improvement among the three efficient EAs.

(2) *Indicator 2: the number of obtaining the minimum average objective value.* As for the average objective value, PS-SEGA can obtain the minimum values (marked in bold in Table 8) in all scenarios compared with PE-DE-HI-NR and PE-ES-HI-HR.

(3) *Indicator 3: the value of $Score$.* From the results of Wilcoxon rank-sum test on the three efficient EAs (Tables 7–9), the $Score$ of PE-DE-HI-NR, PS-SEGA, and PE-ES-HI-HR are 31, 35 and 4, respectively. This means PS-SEGA has the best performance. Therefore, PS-SEGA is the most efficient evolutionary algorithm among the three efficient EAs.

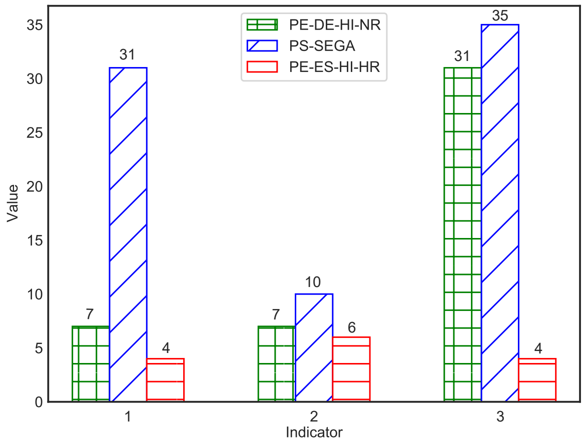



Figure 9. Indicator comparison results of PE-DE-HI-NR, PS-SEGA, and PE-ES-HI-HR. The three indicators in horizontal axis are number from 1 to 3, i.e., 1: the degree of performance improvement, 2: the number of obtaining the minimum average objective value, 3: the value of $Score$.

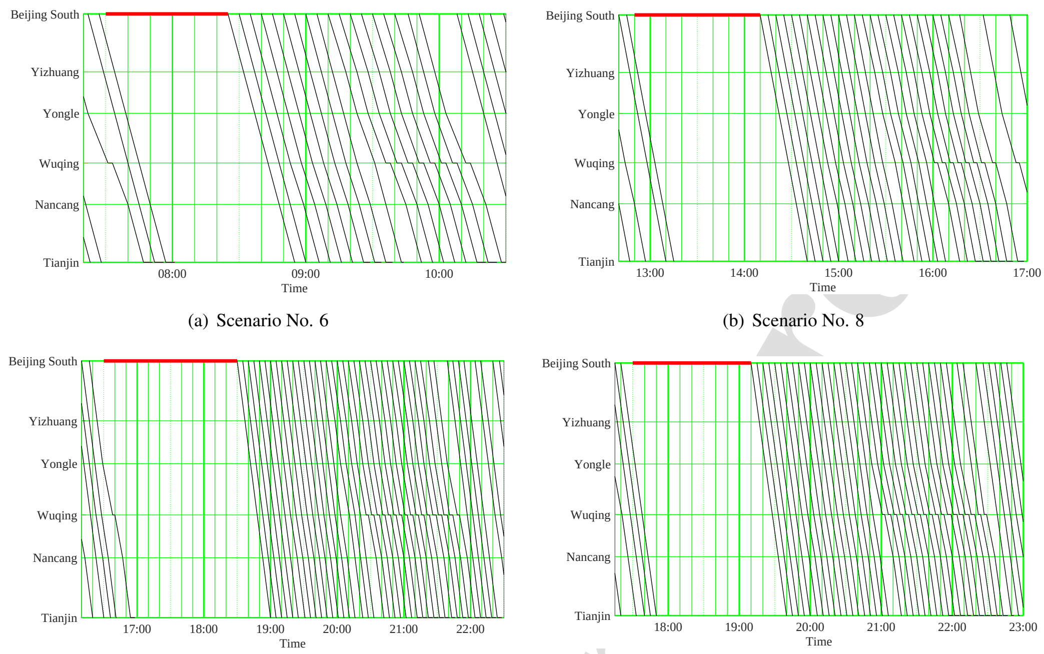
*4.3.5. Rescheduled timetable in PS-SEGA*

To verify the performance of PS-SEGA in reducing the total train delay, CPLEX, FSFS, and EHS are used to address the ten scenarios (Table 3). Table 10 lists the optimized solution (i.e., the best departure sequence at the blockage station), the optimized objective value (OOV, i.e., the total train delay), and the computation time. The best results are also marked in bold.

PS-SEGA can provide real-time solutions in all scenarios (Table 10). Firstly, with the CPLEX solution as a reference, PS-SEGA can provide optimal solutions in the five small disturbance scenarios (No. 1–5). Notably, there is more than one optimal solution to the optimal objective value[1]. Because homogeneous trains have the same priority in the rescheduled timetable, the HSTR problem in numerical experiments is essentially a multi-model optimization problem.

In addition, the computation time in PS-SEGA is about the same as that by CPLEX in the five small disturbance scenarios (No. 1–5). In several scenarios (No. 3, No. 5), PS-SEGA computes faster than CPLEX. If we set a

[1] https://github.com/wrsBJTU/Paper/blob/main/trainDep.csv




(a) Scenario No. 6

(b) Scenario No. 8

(c) Scenario No. 9

(d) Scenario No. 10

Figure 10. Rescheduled timetables in the best departure sequences under four typical scenarios. The solid red line represents the time period under the partial station blockage. The time horizons for the four typical scenarios are [440, 630], [760, 1020], [970, 1350], [1035, 1380].

smaller $NFE_{\max}$, the computation time in PS-SEGA may be less than that in CPLEX in all five small disturbance scenarios. However, with an increase of the problem complexity, CPLEX fails to provide optimal solutions within 10 min (marked as NA in Table 10) in the last five scenarios (No. 6–10). So CPLEX is not suitable for partial station blockage under large disturbances.

Then, we compare the optimized objective values between PS-SEGA and FSFS/EHS to verify the performance of PS-SEGA. Hence, an indicator called relative percentage deviation (*RPD*) is defined to evaluate the effect of minimizing the total train delay of PS-SEGA, i.e.,

$$RPD = (alg - bst)/bst \tag{11}$$

where *alg* is the optimized objective value acquired by FSFS or EHS. The variable *bst* represents the minimum objective value (marked in bold in Table 10) calculated by PS-SEGA. *RPD* also indicates the gap between PS-SEGA and FSFS/EHS. Overall, the *RPD* results using FSFS to address the ten scenarios are 0%, 12.21%, 5.55%, 4.38%, 11.23%, 7.43%, 3.24%, 6.30%, 3.50%, 3.82%. These values indicate the effect of PS-SEGA in reducing the total train delay. The *RPD* results of EHS in the ten scenarios are 0%, 12.21%, 3.14%, 0.36%, 9.79%, 4.66%, 1.58%, 3.73%, 1.54%, 2.02%. Thus, we can conclude that EHS can provide the initial population with a solution close to the optimal value and better than FSFS.

Finally, the best departure sequences in all scenarios are selected by PS-SEGA. With the best departure sequences and the heuristic decoding method, Figs. 10(a)–10(d) present the rescheduled timetable under the four typical scenarios. The solid red line denotes the time period $[t_{\text{start}}, t_{\text{end}})$ under the partial station blockage. We can conclude that the train not dwelling at the intermediate stations will depart earlier from the blockage station than the other trains. In other words, the more train stops at each station along the line, the later the train departs from the blockage station.

Table 10. Comparison results of the optimized solution, optimized objective value, and computation time in the four approaches.

| No. | Approach | Optimized solution | OOV/min | Time/s |
|---|---|---|---|---|
| 1 | PS-SEGA | 0, 1, 2, 3, 4 | **1080**† | 2.548 ± 0.125 |
| | CPLEX | 0, 1, 2, 3, 4 | **1080**† | 1.624 |
| | FSFS | 0, 1, 2, 3, 4 | **1080** | <0.001 |
| | EHS | 0, 1, 2, 3, 4 | **1080**† | <0.001 |
| 2 | PS-SEGA | 1, 2, 0, 3, 4 | **303**† | 2.414 ± 0.292 |
| | CPLEX | 1, 2, 0, 3, 4 | **303**† | 1.478 |
| | FSFS | 0, 1, 2, 3, 4 | 340 | <0.001 |
| | EHS | 0, 1, 2, 3, 4 | 340 | <0.001 |
| 3 | PS-SEGA | 1, 3, 0, 5, 6, 2, 4, 7, 8 | **2036**† | 4.114 ± 0.174 |
| | CPLEX | 1, 3, 0, 5, 6, 4, 2, 7, 8 | **2036**† | 6.185 |
| | FSFS | 0, 1, 2, 3, 4, 5, 6, 7, 8 | 2149 | <0.001 |
| | EHS | 1, 3, 0, 2, 4, 5, 6, 7, 8 | 2100 | <0.001 |
| 4 | PS-SEGA | 1, 2, 3, 0, 4, 5 | **1096**† | 2.880 ± 0.049 |
| | CPLEX | 1, 2, 3, 0, 4, 5 | **1096**† | 2.006 |
| | FSFS | 0, 1, 2, 3, 4, 5 | 1144 | <0.001 |
| | EHS | 1, 2, 0, 3, 4, 5 | 1100 | <0.001 |
| 5 | PS-SEGA | 1, 3, 4, 0, 5, 6, 7, 2 | **1532**† | 3.979 ± 0.033 |
| | CPLEX | 1, 3, 4, 0, 5, 6, 7, 2 | **1532**† | 4.950 |
| | FSFS | 0, 1, 2, 3, 4, 5, 6, 7 | 1704 | <0.001 |
| | EHS | 1, 0, 2, 3, 4, 5, 6, 7 | 1682 | <0.001 |
| 6 | PS-SEGA | 4, 5, 7, 6, 0, 9, 8, 3, 11, 2, 12, 1, 10, 14, 13 | **4725** | 9.249 ± 0.141 |
| | CPLEX | NA | NA | NA |
| | FSFS | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 | 5076 | <0.001 |
| | EHS | 4, 5, 0, 2, 3, 1, 6, 7, 8, 9, 10, 11, 12, 13, 14 | 4945 | <0.001 |
| 7 | PS-SEGA | 0, 3, 5, 4, 1, 6, 7, 2, 8, 9, 10, 11, 12 | **2406** | 5.532 ± 0.160 |
| | CPLEX | NA | NA | NA |
| | FSFS | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 2484 | <0.001 |
| | EHS | 0, 3, 4, 1, 2, 5, 6, 7, 8, 9, 10, 11, 12 | 2444 | <0.001 |
| 8 | PS-SEGA | 2, 5, 7, 3, 9, 1, 10, 8, 12, 13, 14, 15, 16, 11, 4, 18, 19, 20, 17, 21, 0, 6, 22, 23 | **10186** | 11.571 ± 0.256 |
| | CPLEX | NA | NA | NA |
| | FSFS | 0, 1, . . . , 22, 23 | 10828 | <0.001 |
| | EHS | 2, 3, 5, 7, 1, 8, 4, 0, 6, 9, 10, . . . , 22, 23 | 10566 | <0.001 |
| 9 | PS-SEGA | 12, 4, 9, 5, 11, 3, 13, 17, 18, 14, 8, 6, 1, 16, 19, 21, 2, 22, 23, 15, 7, 24, 25, 26, 27, 10, 0, 28, 29, 30, 31, 20, 32, 33, 34, 35 | **22595** | 16.924 ± 0.628 |
| | CPLEX | NA | NA | NA |
| | FSFS | 0, 1, . . . , 34, 35 | 23385 | <0.001 |
| | EHS | 3, 4, 5, 9, 11, 12, 1, 6, 8, 13, 2, 7, 0, 10, 14, 15, . . . , 34, 35 | 22943 | <0.001 |
| 10 | PS-SEGA | 3, 5, 6, 12, 13, 10, 0, 2, 8, 16, 15, 17, 11, 7, 18, 20, 21, 22, 19, 9, 23, 1, 24, 25, 26, 27, 4, 28, 29, 14, 30 | **18636** | 12.975 ± 0.283 |
| | CPLEX | NA | NA | NA |
| | FSFS | 0, 1, . . . , 22, 23 | 19348 | <0.001 |
| | EHS | 3, 5, 6, 12, 0, 2, 7, 8, 10, 11, 1, 9, 4, 13, 14, . . . , 29, 30 | 19012 | <0.001 |

†: Optimal objective value.
NA: CPLEX cannot obtain the optimal solution within 10 min.

## 5. Conclusion

This paper considers the HSTR problem under a partial station blockage. An ILP model is established based on train operation constraints to minimize the total train delay. An efficient evolutionary algorithm called PS-SEGA is proposed to improve the efficiency and quality of the HSTR solution. The Wilcoxon rank-sum test results show that the permutation-based encoding method produces a remarkably better solution than the integer-based encoding method. With ten scenarios under the partial station blockage, a comparative study is performed on DE, SEGA, and ES in different combinations of population initialization methods and restart strategies. Compared with the most efficient improved variants on DE and ES, PS-SEGA has proved to be the most efficient evolutionary algorithm. As for the five large disturbance scenarios in which CPLEX cannot provide the optimal solutions within 10 min, PS-SEGA can provide quasi-optimal HSTR solutions in real time. Finally, the best departure sequence and the corresponding total train delay are provided by PS-SEGA. The corresponding rescheduled timetable is obtained by the heuristic decoding method.

The proposed PS-SEGA could be used as a function of the decision support system to help dispatchers reschedule trains. In our further work, we will construct an efficient approach based on the advantages of PS-SEGA and RL to resolve HSTR. We will also incorporate more actual requirements to optimize the multimodality of HSTR or transform HSTR into a multi-objective optimization problem. Furthermore, we will modify PS-SEGA to apply in the cooperative rescheduling problem of rolling stock and train rescheduling [51], and the integration of train timetable and passenger routing [52, 53]. In addition, the HSTR problem under uncertainty disturbances or disruptions using EAs also be worth investigating [54–56].

### CRediT authorship contribution statement

**Rongsheng Wang:** Conceptualization, Investigation, Validation, Software, Data curation, Formal analysis, Writing - original draft, Writing - review & editing, Visualization. **Qi Zhang:** Conceptualization, Project administration, Supervision. **Xuewu Dai:** Writing - Review & Editing, Validation. **Zhiming Yuan:** Resources, Formal analysis. **Tao Zhang:** Formal analysis, Investigation. **Shuxin Ding:** Methodology, Software. **Yaochu Jin:** Writing - Review & Editing, Investigation.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgements

The research described in the paper was jointly supported by the National Natural Science Foundation of China under Grants 62203468 and U1934220, and by the Foundation of China Academy of Railway Sciences Corporation Limited under Grant 2021YJ043.

### References

[1] Y. H. Min, M. J. Park, S. P. Hong, S. H. Hong, An appraisal of a column-generation-based algorithm for centralized train-conflict resolution on a metropolitan railway network, Transport. Res. B-Meth. 45 (2) (2011) 409–429.
[2] R. M. Lusby, J. T. Haahr, J. Larsen, D. Pisinger, A branch-and-price algorithm for railway rolling stock rescheduling, Transport. Res. B-Meth. 99 (2017) 228–250.
[3] V. Cacchiani, D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, J. Wagenaar, An overview of recovery models and algorithms for real-time railway rescheduling, Transport. Res. B-Meth. 63 (2014) 15–37.
[4] B. Szpigel, Optimal train scheduling on a single track railway, Oper. Res. 72 (1973) 343–351.
[5] P. Pellegrini, G. Marlière, R. Pesenti, J. Rodriguez, RECIFE-MILP: An effective MILP-based heuristic for the real-time railway traffic management problem, IEEE T. Intell. Transp. 16 (5) (2015) 2609–2619.
[6] P. Pellegrini, R. Pesenti, J. Rodriguez, Efficient train re-routing and rescheduling: Valid inequalities and reformulation of RECIFE-MILP, Transport. Res. B-Meth. 120 (2019) 33–48.

[7] S. Zhan, L. G. Kroon, J. Zhao, Q. Peng, A rolling horizon approach to the high speed train rescheduling problem in case of a partial segment blockage, Transport. Res. E-log. 95 (2016) 32–61.
[8] S. Binder, Y. Maknoon, M. Bierlaire, The multi-objective railway timetable rescheduling problem, Transport. Res. C-Emer. 78 (2017) 78–94.
[9] E. Altazin, S. Dauzère-Pérès, F. Ramond, S. Tréfond, A multi-objective optimization-simulation approach for real time rescheduling in dense railway systems, Eur. J. Oper. Res. 286 (2) (2020) 662–672.
[10] Y. Zhu, R. M. Goverde, Railway timetable rescheduling with flexible stopping and flexible short-turning during disruptions, Transport. Res. B-Meth. 123 (2019) 149–181.
[11] A. Mascis, D. Pacciarelli, Job-shop scheduling with blocking and no-wait constraints, Eur. J. Oper. Res. 143 (3) (2002) 498–517.
[12] A. D'Ariano, M. Pranzo, I. A. Hansen, Conflict resolution and train speed coordination for solving real-time timetable perturbations, IEEE T. Intell. Transp. 8 (2) (2007) 208–222.
[13] A. D'Ariano, F. Corman, D. Pacciarelli, M. Pranzo, Reordering and local rerouting strategies to manage train traffic in real time, Transport. Sci. 42 (4) (2008) 405–419.
[14] F. Corman, E. Quaglietta, R. M. Goverde, Automated real-time railway traffic control: An experimental analysis of reliability, resilience and robustness, Transport. Plan. Techn. 41 (4) (2018) 421–447.
[15] P. Xu, F. Corman, Q. Peng, X. Luan, A train rescheduling model integrating speed management during disruptions of high-speed traffic under a quasi-moving block system, Transport. Res. B-Meth. 104 (2017) 638–666.
[16] K. Nitisiri, M. Gen, H. Ohwada, A parallel multi-objective genetic algorithm with learning based mutation for railway scheduling, Comput. Ind. Eng. 130 (2019) 381–394.
[17] M. Wang, L. Wang, X. Xu, Y. Qin, L. Qin, Genetic algorithm-based particle swarm optimization approach to reschedule high-speed railway timetables: A case study in China, J. Adv. Transp. 2019.
[18] A. R. Jafarian-Moghaddam, Elastic train scheduling model, Appl. Soft Comput. 110 (2021) 107627.
[19] Y. Tan, Y. Li, R. Wang, X. Mi, Y. Li, H. Zheng, Y. Ke, Y. Wang, Improving synchronization in high-speed railway and air intermodality: Integrated train timetable rescheduling and passenger flow forecasting, IEEE T. Intell. Transp. 23 (3) (2022) 2651–2667.
[20] Z. Han, B. Han, D. Li, S. Ning, R. Yang, Y. Yin, Train timetabling in rail transit network under uncertain and dynamic demand using advanced and adaptive nsga-ii, Transport. Res. B-Meth. 154 (2021) 65–99.
[21] M. Samà, P. Pellegrini, A. DAriano, J. Rodriguez, D. Pacciarelli, Ant colony optimization for the real-time train routing selection problem, Transport. Res. B-Meth. 85 (2016) 89–108.
[22] J. Eaton, S. Yang, M. Gongora, Ant colony optimization for simulated dynamic multi-objective railway junction rescheduling, IEEE T. Intell. Transp. 18 (11) (2017) 2980–2992.
[23] B. Pascariu, M. Sama, P. Pellegrini, A. DAriano, J. Rodriguez, D. Pacciarelli, Effective train routing selection for real-time traffic management: Improved model and aco parallel computing, Comput. Oper. Res. 145 (2022) 105859.
[24] S. Ding, C. Chen, B. Xin, P. M. Pardalos, A bi-objective load balancing model in a distributed simulation system using NSGA-II and MOPSO approaches, Appl. Soft Comput. 63 (2018) 249–267.
[25] M. Guo, B. Xin, J. Chen, Y. Wang, Multi-agent coalition formation by an efficient genetic algorithm with heuristic initialization and repair strategy, Swarm Evol. Comput. 55 (2020) 100686.
[26] Y. Wang, B. Xin, J. Chen, An adaptive memetic algorithm for the joint allocation of heterogeneous stochastic resources, IEEE Trans. Cybern. 52 (11) (2021) 11526–11538.
[27] G. Gao, Y. Mei, Y.-H. Jia, W. N. Browne, B. Xin, Adaptive coordination ant colony optimization for multipoint dynamic aggregation, IEEE T. Cybernetics 52 (8) (2022) 7362–7376.
[28] L. Cui, X. Liu, S. Lu, Z. Jia, A variable neighborhood search approach for the resource-constrained multi-project collaborative scheduling problem, Appl. Soft Comput. 107 (2021) 107480.
[29] J. Yin, D. Chen, L. Li, Intelligent train operation algorithms for subway by expert system and reinforcement learning, IEEE T. Intell. Transp. 15 (6) (2014) 2561–2571.
[30] Z. Jiang, W. Fan, W. Liu, B. Zhu, J. Gu, Reinforcement learning approach for coordinated passenger inflow control of urban rail transit in peak hours, Transport. Res. C-Emer. 88 (2018) 1–16.
[31] F. Zhu, Z. Yang, F. Lin, Y. Xin, Decentralized cooperative control of multiple energy storage systems in urban railway based on multiagent deep reinforcement learning, IEEE T. Power Electr. 35 (9) (2020) 9368–9379.
[32] W. Liu, S. Su, T. Tang, X. Wang, A DQN-based intelligent control method for heavy haul trains on long steep downhill section, Transport. Res. C-Emer. 129 (2021) 103249.
[33] D. Šemrov, R. Marsetič, M. Žura, L. Todorovski, A. Srdic, Reinforcement learning approach for train rescheduling on a single-track railway, Transport. Res. B-Meth. 86 (2016) 250–267.
[34] H. Khadilkar, A scalable reinforcement learning algorithm for scheduling railway lines, IEEE T. Intell. Transp. 20 (2) (2019) 727–736.
[35] L. Ning, Y. Li, M. Zhou, H. Song, H. Dong, A deep reinforcement learning approach to high-speed train timetable rescheduling under disturbances, in: 2019 IEEE T. Intell. Transp. Conf., IEEE, 2019, pp. 3469–3474.
[36] R. Wang, M. Zhou, Y. Li, Q. Zhang, H. Dong, A timetable rescheduling approach for railway based on Monte Carlo tree search, in: 2019 IEEE T. Intell. Transp. Conf., IEEE, 2019, pp. 3738–3743.
[37] Q. Zhang, J. Jin, Y. Liu, et al., Interim technical specification for intelligent centralized traffic control (CTC) system, China State Railway Group Corporation Limited, TJ/DW208-2019, 2019-9-27.
[38] X. Yang, B. Ning, X. Li, T. Tang, A two-objective timetable optimization model in subway systems, IEEE T. Intell. Transp. 15 (5) (2014) 1913–1921.
[39] G. Cavone, L. Blenkers, T. Van Den Boom, M. Dotoli, C. Seatzu, B. De Schutter, Railway disruption: A bi-level rescheduling algorithm, in: 2019 6th Int. Conf. Control. Decis. Inf. Technol., IEEE, 2019, pp. 54–59.
[40] J. Wang, J. Wang, A new early warning method of train tracking interval based on CTC, IEEE T. Intell. Transp.
[41] S. Venkatraman, G. G. Yen, A generic framework for constrained optimization using genetic algorithms, IEEE T. Evolut. Comput. 9 (4) (2005) 424–435.

[42] H. Zhang, F. Liu, Y. Zhou, Z. Zhang, A hybrid method integrating an elite genetic algorithm with tabu search for the quadratic assignment problem, Inf. Sci. 539 (2020) 347–374.
[43] S. Das, P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE T. Evolut. Comput. 15 (1) (2011) 4–31.
[44] F. Corman, A. DAriano, A. D. Marra, D. Pacciarelli, M. Samà, Integrating train scheduling and delay management in real-time railway traffic control, Transport. Res. E-log. 105 (2017) 213–239.
[45] S. Das, A. Abraham, A. Konar, Automatic clustering using an improved differential evolution algorithm, IEEE T. Syst. Man. Cy. A. 38 (1) (2007) 218–237.
[46] P. Civicioglu, E. Besdok, A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms, Artif. Intell. Rev. 39 (4) (2013) 315–346.
[47] W. Chen, M. Panahi, H. R. Pourghasemi, Performance evaluation of gis-based new ensemble data mining techniques of adaptive neuro-fuzzy inference system (ANFIS) with genetic algorithm (GA), differential evolution (DE), and particle swarm optimization (PSO) for landslide spatial modelling, Catena 157 (2017) 310–324.
[48] H.-G. Beyer, H.-P. Schwefel, Evolution strategies–a comprehensive introduction, Nat. Comput. 1 (1) (2002) 3–52.
[49] J. Jazzbin, Geatpy: the genetic and evolutionary algorithm toolbox with high performance in python (2020).
[50] M. Baioletti, A. Milani, V. Santucci, Algebraic particle swarm optimization for the permutations search space, in: 2017 IEEE Congr. Evol. Comput., IEEE, 2017, pp. 1587–1594.
[51] L. Kang, Y. Xiao, H. Sun, J. Wu, S. Luo, N. Buhigiro, Decisions on train rescheduling and locomotive assignment during the covid-19 outbreak: A case of the beijing-tianjin intercity railway, Decis. Support Syst. 161 (2022) 113600.
[52] F. Corman, Interactions and equilibrium between rescheduling train traffic and routing passengers in microscopic delay management: A game theoretical study, Transport. Sci. 54 (3) (2020) 785–822.
[53] S. Zhan, S. Wong, P. Shang, Q. Peng, J. Xie, S. Lo, Integrated railway timetable rescheduling and dynamic passenger routing during a complete blockage, Transport. Res. B-Meth. 143 (2021) 86–123.
[54] H. Zhang, S. Li, Y. Wang, Y. Wang, L. Yang, Real-time optimization strategy for single-track high-speed train rescheduling with disturbance uncertainties: A scenario-based chance-constrained model predictive control approach, Comput. Oper. Res. 127 (2021) 105135.
[55] Y. Zhu, R. M. Goverde, Dynamic and robust timetable rescheduling for uncertain railway disruptions, J. Rail Transport Pla. 15 (2020) 100196.
[56] S. Ding, C. Chen, Q. Zhang, B. Xin, P. M. Pardalos, Metaheuristics for resource deployment under uncertainty in complex systems, 1st Edition, CRC Press, Boca Raton FL, USA, 2021.

## Highlights

1. A high-speed train rescheduling model with time constraints is established.
2. A permutation-based encoding method is proposed to improve solution efficiency.
3. A heuristic decoding method is developed to eliminate all time constraints.
4. Problem-specific knowledge is designed to provide an efficient rescheduling solution.
5. Ten real-world cases are used to verify the effectiveness of the proposed algorithm.

Rongsheng Wang: Conceptualization, Investigation, Validation, Software, Data curation, Formal analysis, Writing - original draft, Writing - review & editing, Visualization.

Qi Zhang: Conceptualization, Project administration, Supervision.

Xuewu Dai: Writing - Review & Editing, Validation.

Zhiming Yuan: Resources, Formal analysis.

Tao Zhang: Formal analysis, Investigation.

Shuxin Ding: Methodology, Software.

Yaochu Jin: Writing - Review & Editing, Investigation.

## Declaration of Interest Statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.