https://doi.org/10.20965/jaciii.2022.p0407

Paper:

A Memetic Algorithm for High-Speed Railway Train Timetable Rescheduling

Shuxin Ding*1,*2, Tao Zhang*1,*2,†, Ziyuan Liu*3, Rongsheng Wang*1,*2,*4, Sai Lu*5,*6, Bin Xin*5,*6, and Zhiming Yuan*1,*2

*1 Signal and Communication Research Institute, China Academy of Railway Sciences Co., Ltd.
No.2 Daliushu Road, Haidian District, Beijing 100081, China
E-mail: 13701193534@139.com
*2 Train Operation Control Laboratory for High-Speed Railway,
National Engineering Research Center of System Technology for High-Speed Railway and Urban Rail Transit,
China Academy of Railway Sciences Co., Ltd.
No.2 Daliushu Road, Haidian District, Beijing 100081, China
*3 China Academy of Railway Sciences Co., Ltd.
No.2 Daliushu Road, Haidian District, Beijing 100081, China
*4 Postgraduate Department, China Academy of Railway Sciences
No.2 Daliushu Road, Haidian District, Beijing 100081, China
*5 School of Automation, Beijing Institute of Technology
No.5 Zhongguancun South Street, Haidian District, Beijing 100081, China
*6 State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology
No.5 Zhongguancun South Street, Haidian District, Beijing 100081, China
† Corresponding author

[Received December 8, 2021; accepted March 8, 2022]

This study addresses a high-speed railway train timetable rescheduling (TTR) problem with a complete blockage at the station and train operation constraints. The problem is formulated as a mixed-integer linear programming (MILP) model that minimizes the weighted sum of the total delay time of trains. A memetic algorithm (MA) is proposed, and the individual of MA is represented as a permutation of trains' departure order at the disrupted station. The individual is decoded to a feasible schedule of the trains using a rule-based method to allocate the running time in sections and dwell time at stations. Consequently, the original problem is reformulated as an unconstrained problem. Several permutation-based operators are involved, including crossover, mutation, and local search. A restart strategy was employed to maintain the the population diversity. The proposed MA was compared with the first-scheduled-first-served (FSFS) algorithm and other state-of-the-art evolutionary algorithms. The experimental results demonstrate the superiority of MA in solving the TTR through permutation-based optimization in terms of constraint handling, solution quality, and computation time.

Keywords: high-speed railway, train timetable rescheduling, disruptions, memetic algorithm, combinatorial optimization

1. Introduction

High-speed railway (HSR) plays an important role in medium-to-long-distance transportation services in China. HSR operates according to the prescribed timetable. However, HSR may face inevitable emergencies such as infrastructure failure, train failure, and natural disasters [1]. The operations of the train may be disturbed or disrupted by delays. Therefore, train timetable rescheduling (TTR) is required for trains to recover their regular operation.

Various studies have been conducted on the TTR problem, which has been proven to be NP-hard [2, 3]. In most studies, a mixed-integer linear programming (MILP) model was adopted, and the CPLEX solver was used to obtain solutions. An MILP model was proposed in [4] to deal with the real-time rescheduling of a timetable in the case of a complete blockage in a railway segment. However, when the scale of the problem increases, using the CPLEX solver becomes time consuming, which may exceed the time limit.

Metaheuristics are typically used to solve NP-hard problems [5]. Near-optimal solutions were obtained within a limited time. Genetic algorithm-based particle swarm optimization has been used to reschedule HSR timetables under primary delays [6]. Meng et al. [7] considered train rescheduling with track assignment, and proposed an artificial bee colony algorithm to solve this problem. The departure and arrival times of the trains are used for the solution representation. However, this may result in constraint-violated solutions. Few related works have considered determining the order of trains to determine the train timetable. Wang et al. [8] adjusted train departure sequences based on a Monte Carlo tree search. Wang and Wang [9] proposed an effective estimation of the distribution algorithm (EDA) to solve the multi-track train scheduling problem. The permutation of the train priority was obtained. Ding et al. [10] used several metaheuristics to solve the TTR problem using permutation-based combinatorial optimization. However, the proposed metaheuristics are not well designed for permutation-based optimization problems.

The memetic algorithm (MA) is a population-based metaheuristic that combines evolutionary algorithms (EA) and local search techniques [11]. It has been adopted to solve many complex optimization problems [12–15]. The encoding scheme, genetic operator (i.e., selection, crossover, and mutation operators), and local search strategy are important for MA performance. Some of these are specially designed for permutation-based optimization problems.

The main contributions of this study are summarized as follows: first, a high-speed railway train timetable rescheduling problem with a complete station blockage is proposed and modeled as an MILP problem. Second, an effective permutation encoding method was proposed for the TTR problem, and a rule-based decoding method was designed to obtain a new schedule. These encoding and decoding methods can manage all the constraints and guarantee the feasibility of the solution. Subsequently, an MA is proposed to solve the permutation-based optimization problem with various permutation-based operators. A local search operator was developed to exploit the neighborhood of the best current individual, and a restart strategy was used to maintain the diversity of the population. Finally, the experimental results show that the proposed MA can efficiently solve most test instances compared to state-of-the-art algorithms.

The remainder of this paper is organized as follows. The TTR problem is described in Section 2. Section 3 presents a memetic algorithm for solving the TTR. The performance of the proposed algorithms is evaluated in Section 4. Finally, conclusions and future work are presented in Section 5.

2. Problem Formulation

Punctuality is an important factor in railway operation. However, in the case of disruptions, the railway system may fall into disorder. Trains may not be able to arrive or depart at the stations.

In this section, we introduce the MILP model to formulate the TTR problem. We need to determine the new arrival and departure times of trains at stations to recover railway operations.

There are seven assumptions. (1) All trains should follow their original schedules before disruption occurs. (2) No trains are canceled in the train timetable rescheduling problem. (3) A macroscopic model was presented without considering signaling systems and station capacity. (4) The disruption considered was complete blockage at the first station. All the affected trains departed after the disruption ended. (5) There is only one disruption with a known duration. (6) Train reordering is not allowed, except for departing trains at the first station. (7) Running and dwell time supplements were not considered in the original timetable.

2.1. Indices

i, j: index of the train, $i, j \in T$.
s: index of the station, $s \in S$.
$(s, s+1)$: index of the section between stations s and $s+1$, $(s, s+1) \in K$.
s^*: index of the disrupted station, $s^* \in S$.
$O(i)$ and $D(i)$: index of the origin and destination stations of train i, respectively.

2.2. Parameters

T: the set of trains.
S: the set of stations.
K: the set of sections.
$T^a_{i,s}$: arrival time of train i at station s in the original schedule.
$T^d_{i,s}$: departure time of train i at station s in the original schedule.
$d_{i,s}$: minimum dwell time at station s for train i.
$Y_{i,s}$: train stop indicator in the original schedule, 1 if train i stops at station s; 0 otherwise.
$r^{min}_{i,(s,s+1)}$: minimum running time at section $(s, s+1)$ for train i.
$r^s_{i,(s,s+1)}$: additional time caused by starting train i in section $(s, s+1)$.
$r^e_{i,(s,s+1)}$: additional time caused by stopping train i in section $(s, s+1)$.
$h_{(s,s+1)}$: minimal headway between two consecutive trains in the same direction in section $(s, s+1)$.
w_i: weight value for train i.
M: a large positive number.
H^s_{dis}: start time of disruption.
D_{dis}: duration of disruption.

2.3. Decision Variables

$t^a_{i,s}$: actual arrival time of train i at station s.
$t^d_{i,s}$: actual departure time of train i at station s.
$q_{i,j,(s,s+1)}$: actual traversing order, 1 if train i traverses section $(s, s+1)$ before train j; 0 otherwise.
$y_{i,s}$: actual train stop indicator, 1 if train i stops at station s; 0 otherwise.

2.4. Mathematical Formulation

The optimization model for the TTR problem is an MILP model, which can be formulated as follows:

$$\min\ F=\sum_{i\in T}\sum_{s\in S}w_i\left(t^a_{i,s}-T^a_{i,s}+t^d_{i,s}-T^d_{i,s}\right) \tag{1}$$

$$\text{s.t.}\quad t^d_{i,s}-t^a_{i,s}\geq d_{i,s},\ \forall i\in T;s\in S, \tag{2}$$

$$t^a_{i,s+1}-t^d_{i,s} \geq r^{min}_{i,(s,s+1)}+r^s_{i,(s,s+1)}y_{i,s}+r^e_{i,(s,s+1)}y_{i,s+1},\ \forall i\in T;s\in S\backslash D(i), \tag{3}$$

$$t^d_{j,s}-t^d_{i,s} \geq h_{(s,s+1)}q_{i,j,(s,s+1)}-M\left(1-q_{i,j,(s,s+1)}\right),\ \forall i,j\in T;i\neq j;s\in S\backslash D(i), \tag{4}$$

$$t^a_{j,s+1}-t^a_{i,s+1} \geq h_{(s,s+1)}q_{i,j,(s,s+1)}-M\left(1-q_{i,j,(s,s+1)}\right),\ \forall i,j\in T;i\neq j;s\in S\backslash D(i), \tag{5}$$

$$q_{i,j,(s,s+1)}+q_{j,i,(s,s+1)}=1,\ \forall i,j\in I;i\neq j;s\in S\setminus D(i), \tag{6}$$

$$t^a_{i,s}=T^a_{i,s},\ \forall i\in T;s\in S:T^a_{i,s}\leq H^s_{dis}, \tag{7}$$

$$t^d_{i,s}=T^d_{i,s},\ \forall i\in T;s\in S:T^d_{i,s}\leq H^s_{dis}, \tag{8}$$

$$t^a_{i,s^*}\geq H^s_{dis}+D_{dis},\ \forall i\in T:H^s_{dis}\leq T^a_{i,s^*}\leq H^s_{dis}+D_{dis}, \tag{9}$$

$$t^a_{i,O(i)}=t^d_{i,O(i)},\ \forall i\in T, \tag{10}$$

$$t^a_{i,s}\geq T^a_{i,s},\ \forall i\in T;s\in S, \tag{11}$$

$$t^d_{i,s}\geq T^d_{i,s},\ \forall i\in T;s\in S, \tag{12}$$

$$q_{i,j,(O(i),O(i)+1)}=q_{i,j,(s,s+1)},\ \forall i,j\in T;i\neq j;s\in S\backslash\{O(i),D(i)\}, \tag{13}$$

$$y_{i,s}\leq t^d_{i,s}-t^a_{i,s},\ \forall i\in T;s\in S\backslash\{O(i),D(i)\}, \tag{14}$$

$$y_{i,s}\geq\frac{t^d_{i,s}-t^a_{i,s}}{M},\ \forall i\in T;s\in S\backslash\{O(i),D(i)\}, \tag{15}$$

$$y_{i,s}\geq Y_{i,s},\ \forall i\in T;s\in S\backslash\{O(i),D(i)\}, \tag{16}$$

$$y_{i,s}=Y_{i,s},\ \forall i\in T;s\in\{O(i),D(i)\}, \tag{17}$$

$$t^a_{i,s},t^d_{i,s}\geq 0,\ \forall i\in T;s\in S, \tag{18}$$

$$q_{i,j,(s,s+1)}\in\{0,1\},\ \forall i,j\in T;i\neq j;s\in S\backslash D(i), \tag{19}$$

$$y_{i,s}\in\{0,1\},\ \forall i,j\in T;i\neq j;s\in S, \tag{20}$$

where Eq. (1) minimizes the weighted sum of the total delay time, including the delay arrival and departure times of each train at all stations. Eq. (2) represents the minimum dwelling time constraint. Eq. (3) represents the minimum running time constraint. Eqs. (4) and (5) are the headway constraints for the departure and arrival headways, respectively. Eq. (6) is the traverse-order constraint of two trains in a section, which means that either train i traverses section $(s,s+1)$ before train j or later than train j. Eqs. (7) and (8) guarantee that the arrival and departure times for the unaffected trains are equal to the original timetable. Eq. (9) guarantees that no trains arrive at the stations during disruption. Eq. (10) indicates that the arrival and departure times are the same at the origin station. Eqs. (11) and (12) are the timetable constraints that restrict trains from arriving and departing from stations before their original arrival and departure times, respectively. Eq. (13) guarantees that the actual traversing orders of all trains are equal to those in the first section. Eqs. (14)–(17) are the train stop-indicator constraints. Eqs. (18)–(20) restrict the decision variables to real and binary numbers.

Algorithm 1 The memetic algorithm for TTR

Input: original timetable information; disruption information; set of affected trains T_{dis}.
Output: the actual arrival time $t^a_{i,s}$ and departure time $t^d_{i,s}$; the total delay time F.

```
Generate the initial population pop randomly.
Set NFE = |pop|.
while NFE < MaxFEs do
    Select parent individuals through roulette wheel selection.
    Update pop through modified order crossover.
    Update pop through swap mutation.
    Update NFE according to the number of individuals for mutation.
    Merge the new populations with the original ones and obtain the best individuals according to the population size.
    Update pop through local search using Algorithm 3.
    Update NFE according to the iterations of the local search.
    if the number of different individuals in the population pop is less than the predefined threshold σ then
        Regenerate the population pop randomly.
        NFE = NFE + |pop|.
    end if
end while
Find the best individual p through the evolution process in pop and decode it using Algorithm 2.
return t^a_{i,s}, t^d_{i,s}, and F.
```

3. Memetic Algorithm for TTR

Because the TTR problem is NP-hard, there is no polynomial-time algorithm to obtain an exact solution. In this section, an MA is presented to solve TTR. First, encoding and decoding were introduced to transform the original MILP problem into a permutation-based combinatorial optimization problem without constraints. Subsequently, the proposed MA randomly generates the initial population as a set of permutations. The population is iteratively updated by crossover, mutation, local search, and restart operators. MA adopts various search methodologies, including population-based and local search techniques [11]. The evolutionary process is similar to that of the genetic algorithm (GA) for population-based searches, while a local search is developed. When there is no improvement in the population, a population restart will be performed. The MA process is shown in **Algorithm 1**.

3.1. Solution Representation

For TTR, most studies used a real-coded encoding scheme. The arrival and departure times were used as the solution. However, it is easy to obtain constraint violations during the evolutionary computation. Suppose that the traversing order of trains in each section is determined. In this case, we only need to determine the arrival and departure times that satisfy the operation constraints, for example, dwelling time, running time, headway constraints, etc. In this section, we propose a permutation-based encoding method to solve TTR. The problem of determining the traversing order of trains is unconstrained, which is easier than the original MILP problem. The integer number in the solution determines the rescheduling order of trains. For example, a solution $\mathbf{p} = (1,2,4,3,5)$ represents the order of five trains, where train 4 is scheduled before train 3. The orders of the other trains remained the same. Before disruption occurs, trains follow their original schedules, and the set of affected trains T_{dis} can be determined if the arrival time at the first station is after H^s_{dis}. Therefore, $|T_{dis}|$ is the dimension of the permutation-based optimization problem. For the problem using a real-coded encoding scheme, the dimension is $2 \cdot |T_{dis}| \cdot |S|$. The dimension of the problem using a permutation-based encoding scheme has been significantly decreased compared with that using a real-coded encoding scheme. Moreover, the search range for each element also decreased. It decreased from 1440 (minutes of one day) to $|T_{dis}|$ (number of affected trains).

We obtained the actual arrival and departure times through the decoding procedure shown in **Algorithm 2** for a permutation-encoded solution $\mathbf{p} = (p_1, p_2, \ldots, p_{|T|})$. In addition, the feasibility of the solution decoded from the permutation can be guaranteed because a constraint-handling technique is applied during encoding and decoding. It follows the rule that trains may arrive and depart at stations once they are allowed as soon as possible. To better illustrate the constraint-handling process, some of the constraints are described in **Fig. 1**. For example, the minimum running time constraint determines the arrival time. The minimum dwelling time constraint determines the departure time. Headway constraint determines both arrival and departure times. Trains should depart after disruption ends.

Algorithm 2 Decoding procedure

Input: original timetable information; disruption information; set of affected trains T_{dis}; scheduling order of trains $\mathbf{p} = [p_i]_{1\times|T|}$.

Output: the actual arrival time $t^a_{i,s}$ and departure time $t^d_{i,s}$.

1: **for** $i = 1$ to $|T| - |T_{dis}|$ **do**
2:   **for** $s = O(i)$ to $D(i)$ **do**
3:     $t^a_{i,s} = T^a_{i,s}$; $t^d_{i,s} = T^d_{i,s}$.
4:   **end for**
5: **end for**
6: **for** $i = |T| - |T_{dis}| + 1$ to $|T|$ **do**
7:   **if** $i = |T| - |T_{dis}| + 1$ **then**
8:     $t^a_{p_i,O(p_i)} = H^s_{dis} + D_{dis}$.
9:     $t^d_{p_i,O(p_i)} = t^a_{p_i,O(p_i)}$.
10:   **else**
11:     $t^a_{p_i,O(p_i)} = \max\left(t^a_{p_{i-1},O(p_{i-1})} + h_{(O(p_i),O(p_i)+1)}, T^a_{p_i,O(p_i)}\right)$.
12:     $t^d_{p_i,O(p_i)} = \max\left(t^a_{p_i,O(p_i)} + d_{p_i,O(p_i)}, T^d_{p_i,O(p_i)}\right)$.
13:   **end if**
14:   $y_{p_i,O(p_i)} = Y_{p_i,O(p_i)}$.
15:   **for** $s = O(i) + 1$ to $D(i)$ **do**
16:     $y_{p_i,s} = Y_{p_i,s}$.
17:     $t^a_{p_i,s} = \max\left(t^d_{p_i,s-1} + r^{min}_{p_i,(s-1,s)} + y_{p_i,s-1} r^s_{p_i,(s-1,s)} + y_{p_i,s} r^e_{p_i,(s-1,s)}, T^a_{p_i,s}\right)$.
18:     $t^a_{p_i,s} = \max\left(t^a_{p_i,s}, t^a_{p_{i-1},s} + h_{(s-1,s)}\right)$.
19:     $t^d_{p_i,s} = \max\left(t^a_{p_i,s} + d_{p_i,s}, T^d_{p_i,s}\right)$.
20:     **if** $s < D(p_i)$ **then**
21:       $t^d_{p_i,s} = \max\left(t^d_{p_i,s}, t^d_{p_{i-1},s} + h_{(s,s+1)}\right)$.
22:       **if** $\operatorname{sgn}\left(t^d_{p_i,s} - t^a_{p_i,s}\right) > y_{p_i,s}$ **then**
23:         $t^a_{p_i,s} = \min\left(t^d_{p_i,s-1} + r^{min}_{p_i,(s-1,s)} + y_{p_i,s-1} r^s_{p_i,(s-1,s)} + r^e_{p_i,(s-1,s)}, t^d_{p_i,s}\right)$.
24:         $y_{p_i,s} = \operatorname{sgn}\left(t^d_{p_i,s} - t^a_{p_i,s}\right)$.
25:       **end if**
26:     **end if**
27:   **end for**
28: **end for**
29: **return** $t^a_{i,s}$ and $t^d_{i,s}$.

Remark 1: In **Algorithm 2**, all constraints for the TTR are satisfied. In line 22, the condition is satisfied when an additional stop may be added at station s. This may add to the total running time for the section $(s-1,s)$ because of the additional time caused by stopping. Therefore, the arrival times must be updated. If the arrival time is larger than the departure time, the stop is canceled, and the arrival time is set to the departure time.

3.2. Selection Operator

The operator used for selection was the roulette wheel selection. It is typically used in GA. The individuals were selected according to their fitness values. Because this is a problem for minimization, the probabilities of the individuals are set according to the exponential of the negative fitness values.

3.3. Crossover Operator

Modified order crossover (MOC) is adopted for the proposed MA [16]. It is designed for permutation-based combinatorial optimization problems, such as the traveling salesman problem. The MOC operator randomly selects a crossover point to divide both parent individuals p_1 and p_2 into left and right strings of the same length. Then, the order of the right string p_1 is used to change the order of the positions in p_2 and vice versa. **Fig. 2** shows

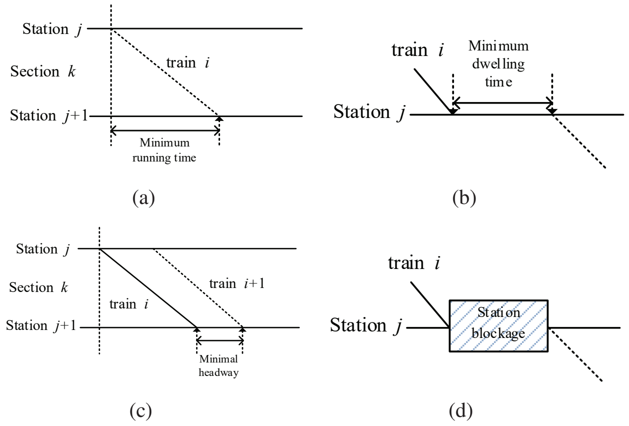


Fig. 1. Determine arrival and departure time in the decoding procedure. (a) Minimum running time constraints. (b) Minimum dwelling time constraints. (c) Headway constraints. (d) Depart after disruption ends.

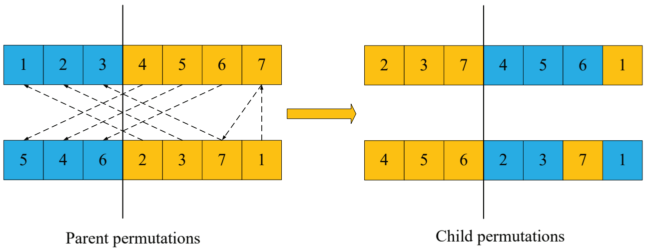


Fig. 2. Modified order crossover.

an example of MOC. The MOC operator is employed to produce the offsprings of two parent individuals based on the crossover rate p_c.

3.4. Mutation Operator

The mutation was conducted after the crossover. It should be mentioned that not all newly obtained individuals were mutated. This is based on the mutation rate p_m. The mutation operator helps to maintain the diversity of the population by changing some of the individuals in the current population. The swap operator is chosen for mutation. We randomly selected two positions in an individual and swapped them to obtain a new permutation.

3.5. Survivor Selection

The survivor selection operator selects the fittest individuals that remain in the population. Truncation selection selects the top solutions with population size based on the objective values.

3.6. Local Search

Local search is an important process in MA that helps maintain the tradeoff between exploration and exploration. The best individual in the population was used for the local search. There are L iterations for the local search. The best individual obtained in the local search process replaces the worst individual in the population. The local search process is shown in **Algorithm 3**.

Algorithm 3 Local search

Input: current population *pop*; best individual p_{best}; number of iterations of local search L; fitness function of the total delay time F.
Output: updated population *pop*.

for $i = 1$ to L **do**
  Perform swap operator on the best individual p_{best} to obtain a new individual $p(i)$.
  Calculate the objective value $F(i)$ for new individual $p(i)$.
end for
Find the best individual $p(i_{best})$ based on $F(i)$.
Replace the worst one in the current population *pop* by $p(i_{best})$.
return *pop*.

3.7. Restart Strategy

During the evolutionary process of MA, the population converges to similar populations, which significantly decreases its diversity. Consequently, it was difficult to generate new solutions. A restart strategy is employed to reduce wasted time and improve the diversity of the population. When the population is updated, the objective values of the new individuals are calculated. If the number of different objective values in the new population is below a predefined threshold σ, the entire population is reinitialized randomly.

4. Computational Experiments

This section presents an investigation of the performance of the proposed algorithms. First, we present the test instances for the TTR. Then, we solved the problem using different methods, including exact solutions using CPLEX. All experiments were carried out on a PC with an Intel Xeon Gold 5218 CPU 2.30 GHz and 32 GB of internal memory. Exact solutions for TTR problems were implemented in MATLAB R2019b using YALMIP as the modelling language and CPLEX 12.10, with default parameter settings [17]. Other algorithms for TTR problems were implemented using MATLAB R2019b.

4.1. Test Instances for TTR

Owing to the lack of benchmark instances with disruptions for TTR in the literature, we first developed the test instances. The Beijing–Tianjin intercity railway timetable from Beijing South to Tianjin was considered in this study. There are altogether six stations and five sections. 40 trains downstream from 6:00 to 12:00 hrs are considered for the railway timetable. The minimum dwell time for train stops at stations was set to 2 min and there

Table 1. The minimum running time in each section between two stations.

No.	Section	Time [min]
1	Beijing South–Yizhuang	5
2	Yizhuang–Yongle	5
3	Yongle–Wuqin	6
4	Wuqin–Nancang	5
5	Nancang–Tianjin	5

Table 2. Setting of the two basic parameters for the test instances.

No.	$\lvert T\rvert$	D_{dis} [min]	No.	$\lvert T\rvert$	D_{dis} [min]
1, 5	15	30	2, 6	20	50
3, 7	30	70	4, 8	40	90

was no dwell time at the pass-through, origin, and destination stations. The minimum running times for each section are listed in **Table 1**. The additional times required for starting and stopping were set to 2 and 3 min, respectively. The minimal headway was set to 4 min. The start time of the disruption H^s_{dis} was set to 6:40 hrs. s^* was set to 1, which means that disruption occurs at the first station. M was set to 1440 min.

We categorize the generation of w_i into the following two cases:

Case 1: The weight values w_i of the trains are set to 1.

Case 2: The weight values w_i of the trains are generated as uniformly distributed random integers in the range of 1 to 10.

Eight test instances were produced to validate the performance of the algorithms. The first four instances (Nos.1–4) were from Case 1, and the last four instances (Nos.5–8) were from Case 2. The settings of the two basic parameters, $|T|$ and D_{dis}, are listed in **Table 2**. For instances where the number of trains $|T|$ is less than 40, for example, instance No.1, the first train is the same train starting at 6:00 hrs. We did not need to adjust the schedule of all 40 trains when the duration of the disruption was only 30 min. Consequently, $|T|$ for different instances was generated according to the duration of the disruption D_{dis}.

4.2. Algorithms for Comparison

To evaluate the performance of the proposed MA, we compared the proposed MA with the following six algorithms: first-scheduled-first-served (FSFS), dual-model estimation of distribution algorithm (DM-EDA) [18], self-adaptive differential evolution algorithm (SaDE), comprehensive learning particle swarm optimizer (CLPSO) [19], covariance matrix adaptation evolution strategy (CMA-ES) [20], and GA [21]. For the GA, we apply the roulette wheel selection, modified order crossover, and swap mutation, which is similar to the proposed MA without a local search and restart strategy.

Remark 2: Because SaDE, CLPSO, and CMA-ES are algorithms designed to search in continuous space, the random key algorithm is applied to transform the real-valued vector to a permutation. Given a real-valued vector $(3.5, 2.4, 1.6, 0.5, 4.1)$, the permutation obtained is the ranking of the real-valued vector $(4, 3, 2, 1, 5)$. The range of each element in the real-valued vector is also the vector dimension.

4.3. Parameter Settings

For all the algorithms, the particle/population size was set to $10 \cdot D$, where D is the dimension of the search space. For the DM-EDA, the subpopulation size N_{adv} is set to D, which is 10% of the population size. The learning rates, μ_n and μ_e, were both set to 0.2. The predefined threshold, ε, was set to 0.01. For the CLPSO, the acceleration constant, c, was set to 1.49445. The inertia weight w was selected to linear decreasing from 0.9 to 0.4. For the GA and MA, the crossover rates p_c were both set to 0.9. The mutation rates p_m were both set to 0.05. For the MA, the number of local search iterations L was set to 100. The predefined threshold, σ, was set to 2. Each algorithm was terminated when the maximum number of fitness evaluations ($10000 \cdot D$) was reached (i.e., $MaxFEs = 10000 \cdot D$). The number of independent runs for each algorithm for each instance was set to 20.

Most of the algorithm parameters were kept the same as those in the original papers. Additionally, for the CPLEX solver, the termination time was set to 3600 s.

4.4. Results and Analysis

For the proposed MA, the best settings for the number of local search iterations L and predefined threshold σ for the restart strategy are not determined. To analyze the sensitivity of L and σ, we tested MA in all test instances. **Table 3** shows the results of the MA with different parameters (L, σ). The mean value and standard deviation of MA in 20 independent runs are shown in **Table 3**. In seven instances (Nos.1–7), the MA performed well with different parameters. For most parameters, the standard deviation is zero, which means that the parameters are not sensitive. For instance No.8, the best results are indicated in bold. This shows that the number of local search iterations L is set to 100, and the predefined threshold σ for the restart strategy is set to 2.

Based on the given parameter settings, we compared the proposed MA with six algorithms and CPLEX. **Table 4** shows the results of 20 independent runs for each algorithm, with mean values and standard deviations. The CPLEX runs only once. The best results are indicated in bold.

It can be observed from **Table 4** that the proposed MA outperforms the other methods. In seven instances (Nos.1–7), the results of MA were equal to CPLEX. Among these instances, the results of MA are proven to be optimal in instances Nos.1, 2, 5, 6, and 7. Moreover,

Table 3. Results of MA with different parameters (L, σ).

No.	$(20,2)$	$(20,3)$	$(20,4)$	$(50,2)$	$(50,3)$	$(50,4)$	$(80,2)$	$(80,3)$	$(80,4)$	$(100,2)$	$(100,3)$	$(100,4)$
1	1628.0000	1628.0000	1628.0000	1628.0000	1628.0000	1628.0000	1628.0000	1628.0000	1628.0000	1628.0000	1628.0000	1628.0000
	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
2	3874.0000	3874.0000	3874.0000	3874.0000	3874.0000	3874.0000	3874.0000	3874.0000	3874.0000	3874.0000	3874.0000	3874.0000
	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
3	7268.8000	7268.8000	7271.2000	7268.0000	7268.0000	7268.0000	7268.0000	7268.0000	7268.0000	7268.0000	7268.0000	7268.0000
	(2.4623)	(3.5777)	(5.4445)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
4	12070.3000	12070.3000	12070.6000	12070.0000	12070.0000	12070.0000	12070.0000	12070.0000	12070.0000	12070.0000	12070.0000	12070.0000
	(1.3416)	(1.3416)	(1.8468)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
5	6126.0000	6126.0000	6126.0000	6126.0000	6126.0000	6126.0000	6126.0000	6126.0000	6126.0000	6126.0000	6126.0000	6126.0000
	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
6	14810.0000	14810.0000	14810.0000	14810.0000	14810.0000	14810.0000	14810.0000	14810.0000	14810.0000	14810.0000	14810.0000	14810.0000
	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
7	26889.0000	26890.1000	26923.5500	26872.0000	26872.0000	26872.0000	26872.0000	26872.0000	26872.0000	26872.0000	26872.0000	26872.0000
	(45.3965)	(47.1971)	(68.9137)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)	(0.0000)
8	43156.4000	43161.0000	43228.6000	43124.1000	43125.0000	43129.5000	43123.2000	43136.3000	43123.5000	**43122.6000**	43128.9000	43122.9000
	(47.9104)	(58.1088)	(93.8714)	(2.9362)	(10.7508)	(15.3125)	(2.4623)	(56.9516)	(2.6656)	**(1.8468)**	(28.0936)	(2.1981)

Table 4. Results of the comparison on the objective value of different algorithms.

No.	FSFS	DM-EDA	SaDE	CLPSO	CMA-ES	GA	MA	CPLEX
1	1700.00	**1628.00±0.00**‡	**1628.00±0.00**‡	**1628.00±0.00**‡	**1628.00±0.00**‡	**1628.00±0.00**‡	**1628.00±0.00**‡	1628.00‡
2	3962.00	**3874.00±0.00**‡	**3874.00±0.00**‡	**3874.00±0.00**‡	**3874.00±0.00**‡	**3874.00±0.00**‡	**3874.00±0.00**‡	3874.00‡
3	7616.00	7570.80±34.55	7272.80±7.52	7274.40±7.61	7284.30±0.73	7277.50±8.85	**7268.00±0.00**	7268.00†
4	12554.00	12539.20±55.02	**12070.00±0.00**	12072.10±3.34	12081.70±13.27	12072.60±8.00	**12070.00±0.00**	12070.00†
5	8012.00	6462.00±0.00	**6126.00±0.00**‡	**6126.00±0.00**‡	**6126.00±0.00**‡	**6126.00±0.00**‡	**6126.00±0.00**‡	6126.00‡
6	17606.00	15386.00±0.00	**14810.00±0.00**‡	**14810.00±0.00**‡	14852.80±87.82	14852.80±87.82	**14810.00±0.00**‡	14810.00‡
7	35452.00	31475.05±684.50	26874.60±8.00	26875.30±8.32	27177.00±330.65	27038.70±64.42	**26872.00±0.00**‡	26872.00‡
8	61640.00	59492.10±1055.76	43125.00±10.75	43636.00±157.02	43697.00±599.01	43333.50±253.86	**43122.30±1.34**	43128.00†

† CPLEX stopped after running for one hour.
‡ Optimal value.

for instances Nos.3 and 4, the results of MA are equal to those of CPLEX (stopped within one hour). In instance No.8, the result of MA was better than that of CPLEX (stopped within one hour). **Fig. 3** shows the rescheduled train timetable for instance No.8 obtained by MA with an objective value of 43122. From **Fig. 3**, disrupted trains (dotted lines) with fewer stops are scheduled earlier than trains that stop at Wuqin.

In instances Nos.1 and 2, all algorithms except for FSFS converged to the optimal value. This is because the size of the instance is small, and the algorithms can cover nearly all feasible solutions. As for the FSFS, the order of the trains is kept the same, which means that the original order is not optimal under disruption. SaDE also shows good performance in several instances, but it is inferior to that of MA. This is because the SaDE was not designed for permutation-based optimization. A permutation-based MA with a local search mechanism and a restart strategy is highly effective.

Figures 4 and **5** show the convergence curves of the different algorithms in instances Nos.7 and 8. The curves were magnified in some areas for better visualization. The horizontal and vertical axes represent the number of fitness evaluations and mean of the objective function for 20 runs, respectively. It can be observed from **Figs. 4** and **5** that MA converges faster than the other algorithms at the beginning. In addition, both GA and CMA-ES have a high convergence speed. Finally, the final result of MA was better than those of the other algorithms.

Table 5 shows the running times of FSFS, EAs, and CPLEX. The mean values and standard deviations of 20 independent runs for the EAs are shown. The best results are indicated in bold. The results show that SaDE requires more computation time in small-scale instances, whereas DM-EDA requires more computation time in large-scale instances among the EAs. For FSFS, all instances were solved within 0.01 s. It can be observed that all the instances can be solved within one minute. However, the running time for CPLEX increased significantly with an increase in problem size. For instance No.7, the total running time is approximately 2862 s, and for instances Nos.3, 4, and 8, the total running time is more than 3600 s. This result demonstrates the efficiency of the proposed framework with permutation-based encoding and rule-based decoding methods.

Based on the above performance results, we can see that the proposed MA successfully solved most of the test instances for TTR and showed significant advantages compared with the other algorithms. The main reasons for this are as follows:

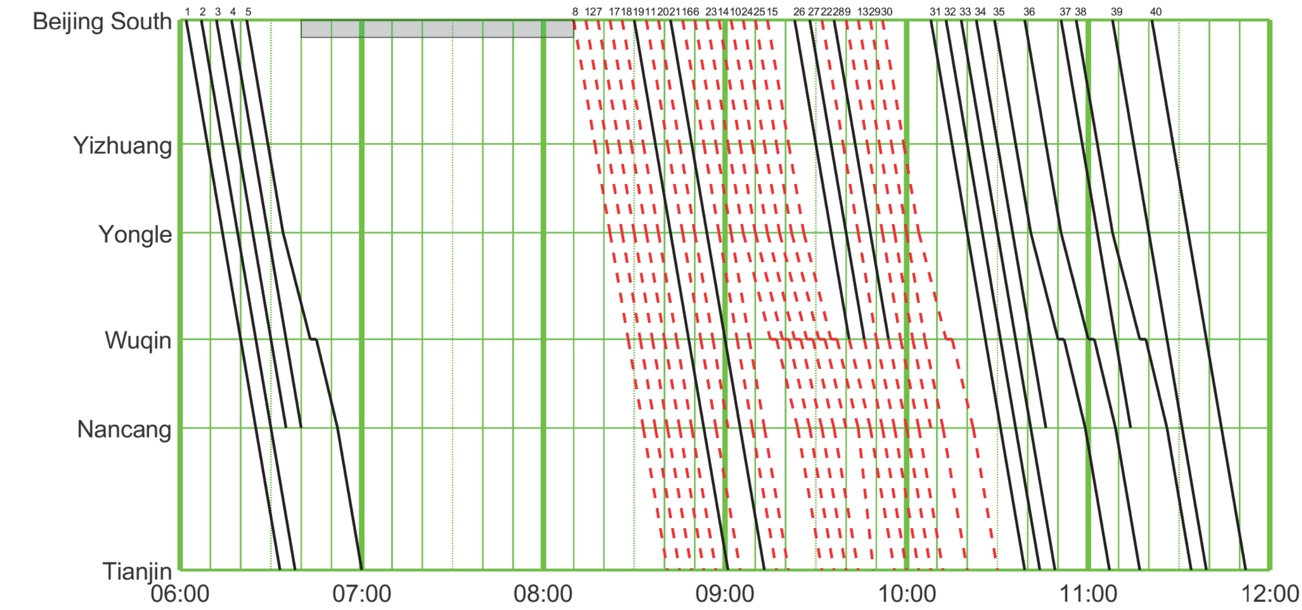


Fig. 3. Rescheduled train timetable for instance No.8.

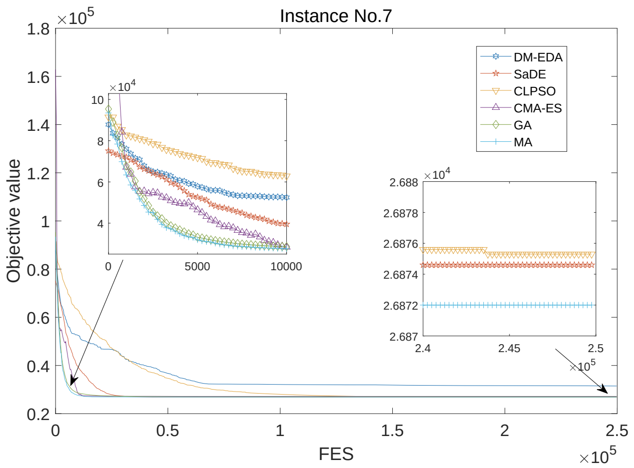


Fig. 4. Convergence curves of different algorithms for instance No.7.

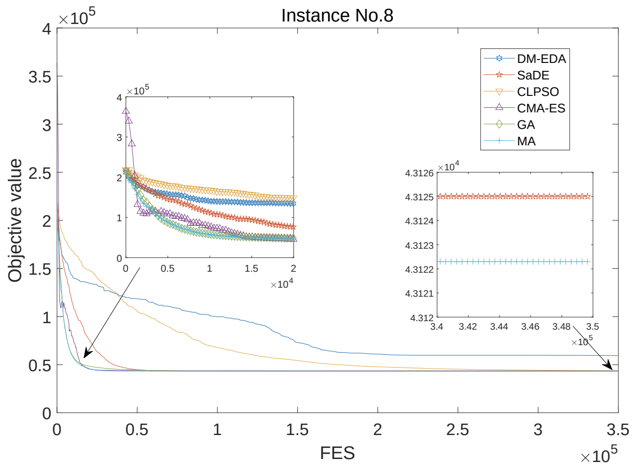


Fig. 5. Convergence curves of different algorithms for instance No.8.

(a) Permutation-based encoding scheme and rule-based decoding method significantly reduce the complexity of the problem. The encoding scheme significantly decreases the search space, and the decoding method guarantees the feasibility of the solution. Therefore, TTR can be solved within a limited computation time.

(b) The proposed MA was designed using a permutation-based encoding scheme. The selection, crossover, and mutation operators are designed for permutation-based combinatorial optimization problems. This makes it more effective for TTR than other algorithms, especially those for continuous space with the random key algorithm to obtain the permutation.

(c) Local search improves the exploration ability of the proposed MA, and the restart strategy improves the diversity of the solution.

5. Conclusion

The high-speed railway TTR problem is formulated as an MILP problem. An MA was proposed to address this problem. A novel encoding and decoding method that transfers the original problem to an unconstrained problem was specially designed for TTR. This avoids numerous ineffective searches in the solution space. With the exception of crossover and mutation operators, the local search strategy and restart strategy are applied to improve the search ability. Based on the testing in eight instances,

Table 5. Runtime performance of different algorithms [sec].

No.	FSFS	DM-EDA	SaDE	CLPSO	CMA-ES	GA	MA	CPLEX
1	0.01	5.74±0.46	8.18±0.65	3.75±0.30	2.24±0.36	4.33±0.20	4.18±0.15	10.39
2	<0.01	10.09±0.69	11.99±0.62	5.57±0.39	3.05±0.27	6.83±0.17	5.89±0.15	64.75
3	<0.01	24.89±0.97	19.91±1.24	11.27±1.80	6.07±1.00	12.98±0.28	11.77±0.24	–
4	<0.01	47.55±1.82	30.01±2.13	17.36±0.65	9.67±0.19	20.46±0.15	19.18±0.38	–
5	0.01	5.12±0.55	8.11±1.17	3.71±0.69	1.88±0.31	4.33±0.13	3.61±0.14	10.55
6	<0.01	10.28±1.29	12.31±2.03	6.06±0.71	2.76±0.12	6.84±0.20	6.05±0.22	30.59
7	<0.01	24.89±0.82	20.10±1.15	11.31±1.33	6.27±1.10	12.87±0.19	11.81±0.20	2861.86
8	<0.01	49.87±3.10	31.19±2.93	17.41±0.82	10.46±1.63	20.55±0.31	19.23±0.31	–

– : CPLEX cannot find optimal value after running for one hour.

the proposed MA outperformed the other algorithms and demonstrated its efficiency compared with CPLEX. The results were obtained within one minute, which is suitable for real-time rescheduling. In the future, we will consider situations with more types of trains (e.g., trains with different prefixes, including G, C, and D) and even reordering at other stations based on the features of the timetable. In addition, considering the uncertainties in a dynamic environment will make the model more practical [22]. The proposed MA can also be improved using a constructive heuristic to obtain good solutions for the initial population [11]. Meanwhile, the multi-objective TTR problem with more optimization objectives using metaheuristics deserves further research [23, 24].

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants U1834211, U1934220, 61822304, and 62088101, and in part by the Foundation of China Academy of Railway Sciences Co., Ltd. (No.2021YJ043).

References:

[1] M. Zhou, H. Dong, X. Liu, H. Zhang, and F.-Y. Wang, "Integrated timetable rescheduling for multidispatching sections of high-speed railways during large-scale disruptions," IEEE Trans. Comput. Soc. Syst., Vol.9, No.2, pp. 366-375, 2021.

[2] V. Cacchiani, D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, and J. Wagenaar, "An overview of recovery models and algorithms for real-time railway rescheduling," Transp. Res. Part B: Methodol., Vol.63, pp. 15-37, 2014.

[3] W. Fang, S. Yang, and X. Yao, "A survey on problem models and solution approaches to rescheduling in railway networks," IEEE Trans. Intell. Transp. Syst., Vol.16, No.6, pp. 2997-3016, 2015.

[4] S. Zhan, L. G. Kroon, L. P. Veelenturf, and J. C. Wagenaar, "Real-time high-speed train rescheduling in case of a complete blockage," Transp. Res. Part B: Methodol., Vol.78, pp. 182-201, 2015.

[5] S. Ding, C. Chen, Q. Zhang, B. Xin, and P. M. Pardalos, "Metaheuristics for Resource Deployment Under Uncertainty in Complex Systems," CRC Press, 2021.

[6] M. Wang, L. Wang, X. Xu, Y. Qin, and L. Qin, "Genetic algorithm-based particle swarm optimization approach to reschedule high-speed railway timetables: A case study in China," J. Adv. Transp., Article No.6090742, 2019.

[7] X. Meng, Y. Wang, W. Xiang, and L. Jia, "An integrated model for train rescheduling and station track assignment," IET Intell. Transp. Syst., Vol.15, Issue 1, pp. 17-30, 2021.

[8] R. Wang, M. Zhou, Y. Li, Q. Zhang, and H. Dong, "A timetable rescheduling approach for railway based on Monte Carlo tree search," Proc. IEEE Intell. Transp. Syst. Conf. (ITSC), pp. 3738-3743, 2019.

[9] S. Wang and L. Wang, "An effective estimation of distribution algorithm for multi-track train scheduling problem," Int. Conf. Intell. Comput. (ICIC2014), Vol.8589, pp. 697-708, 2014.

[10] S. Ding, T. Zhang, R. Wang, C. Zhang, S. Lu, and B. Xin, "A comparative study on evolutionary algorithms for high-speed railway train timetable rescheduling problem," 7th Int. Work. Adv. Comput. Intell. Intell. Informatics (IWACIII2021), Article No.T2-1-6, 2021.

[11] F. Neri and C. Cotta, "Memetic algorithms and memetic computing optimization: A literature review," Swarm Evol. Comput., Vol.2, pp. 1-14, 2012.

[12] Y. Wang, B. Xin, L. Dou, and Z. Peng, "A heuristic initialized memetic algorithm for the joint allocation of heterogeneous stochastic resources," IEEE Congr. Evol. Comput. (CEC), pp. 1929-1936, 2019.

[13] Y. Ding, B. Xin, H. Zhang, and J. Chen, "A memetic algorithm for curvature-constrained path planning of messenger UAV in air-ground coordination," IEEE Int. Conf. Syst. Man, Cybern. (SMC), pp. 1465-1472, 2020.

[14] G. Gao, Y. Mei, B. Xin, Y.-H. Jia, and W. Browne, "A memetic algorithm for the task allocation problem on multi-robot multi-point dynamic aggregation missions," IEEE Congr. Evol. Comput. (CEC), 2020.

[15] Y. Wang, B. Xin, and J. Chen, "An adaptive memetic algorithm for the joint allocation of heterogeneous stochastic resources," IEEE Trans. Cybern., 2021.

[16] A. J. Umbarkar and P. D. Sheth, "Crossover operators in genetic algorithms: A review," ICTACT J. Soft Comput., Vol.6, No.1, pp. 1083-1092, 2015.

[17] J. Lofberg, "YALMIP: a toolbox for modeling and optimization in MATLAB," Int. Conf. on Robotics and Automation, pp. 284-289, 2004.

[18] S. Lu and B. Xin, "Tabu-model-based estimation of distribution algorithm framework for permutation optimization problems," 9th Int. Symp. Comput. Intell. Ind. Appl. (ISCIIA2020), Article No.1A2-2-2, 2020.

[19] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," IEEE Trans. Evol. Comput., Vol.10, No.3, pp. 281-295, 2006.

[20] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation," Proc. IEEE Int. Conf. Evol. Comput., pp. 312-317, 1996.

[21] S. Mirjalili, "Genetic Algorithm," "Evolutionary algorithms and neural networks," pp. 43-55, Springer, 2019.

[22] S. Ding, R. Wang, X. Zhou, Y. Ren, and K. Huang, "High-speed railway train timetable rescheduling in case of a stochastic section blockage," Chinese Autom. Congr. (CAC), pp. 4322-4327, 2021.

[23] S. Ding, C. Chen, B. Xin, and P. M. Pardalos, "A bi-objective load balancing model in a distributed simulation system using NSGA-II and MOPSO approaches," Appl. Soft Comput., Vol.63, pp. 249-267, 2018.

[24] W. Zheng, Y. Tan, M. Gao, W. Jia, and Q. Wang, "A modification of MOEA/D for solving multi-objective optimization problems," J. Adv. Comput. Intell. Intell. Inform., Vol.22, No.2, pp. 214-223, 2018.

Name:
Shuxin Ding

Affiliation:
Associate Researcher, Signal and Communication Research Institute, China Academy of Railway Sciences Co., Ltd.

Address:
No.2 Daliushu Road, Haidian District, Beijing 100081, China

Brief Biographical History:
2019- Assistant Researcher, Signal and Communication Research Institute, China Academy of Railway Sciences Co., Ltd.
2021- Associate Researcher, Signal and Communication Research Institute, China Academy of Railway Sciences Co., Ltd.

Main Works:
• S. Ding, C. Chen, B. Xin, and P. M. Pardalos, "A bi-objective load balancing model in a distributed simulation system using NSGA-II and MOPSO approaches," Applied Soft Computing, Vol.63, pp. 249-267, 2018.
• S. Ding, C. Chen, Q. Zhang, B. Xin, and P. M. Pardalos, "Metaheuristics for Resource Deployment under Uncertainty in Complex Systems," CRC Press, 2021.

Membership in Academic Societies:
• Chinese Association of Automation (CAA)
• The Institute of Electrical and Electronics Engineers (IEEE)

Name:
Tao Zhang

Affiliation:
Professor, Signal and Communication Research Institute, China Academy of Railway Sciences Co., Ltd.

Address:
No.2 Daliushu Road, Haidian District, Beijing 100081, China

Brief Biographical History:
2020- Professor, Signal and Communication Research Institute, China Academy of Railway Sciences Co., Ltd.

Main Works:
• T. Zhang, "High speed train rescheduling based on a cooperative particle swarm optimization algorithm," J. of Information and Computational Science, Vol.11, No.17, pp. 6337-6347, 2014.

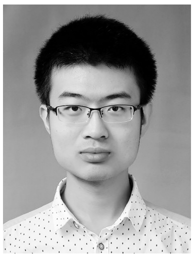
Name:
Ziyuan Liu

Affiliation:
Associate Researcher, China Academy of Railway Sciences Co., Ltd.

Address:
No.2 Daliushu Road, Haidian District, Beijing 100081, China

Brief Biographical History:
2008-2012 Research Assistant, Signal and Communication Research Institute, China Academy of Railway Sciences
2017- Associate Researcher, China Academy of Railway Sciences Co., Ltd.

Main Works:
• Z. Liu, "Optimization of computer-aided decision-making system for railroad traffic dispatching command," Computer-Aided Design & Applications, Vol.19, No.S4, pp. 123-134, 2022.



Name:
Rongsheng Wang

Affiliation:
Ph.D. Student, Postgraduate Department, China Academy of Railway Sciences

Address:
No.2 Daliushu Road, Haidian District, Beijing 100081, China

Brief Biographical History:
2013-2017 B.S. Student, School of Information Science and Technology, Southwest Jiaotong University
2017-2020 M.S. Student, School of Electronic and Information Engineering, Beijing Jiaotong University
2020- Ph.D. Student, Postgraduate Department, China Academy of Railway Sciences

Main Works:
• R. Wang, M. Zhou, Y. Li, Q. Zhang, and H. Dong, "A timetable rescheduling approach for railway based on Monte Carlo tree search," Proc. IEEE Intell. Transp. Syst. Conf. (ITSC), pp. 3738-3743, 2019.

Name:
Sai Lu

Affiliation:
Ph.D. Student, School of Automation, Beijing Institute of Technology

Address:
No.5 Zhongguancun South Street, Haidian District, Beijing 100081, China

Brief Biographical History:
2013-2017 B.S. Student, School of Automation, Harbin Engineering University
2017-2019 M.S. Student, School of Automation, Beijing Institute of Technology
2019- Ph.D. Student, School of Automation, Beijing Institute of Technology

Main Works:
- S. Lu, B. Xin, L. Dou, and L. Wang, "A comparative study on evolutionary algorithms for the agent routing problem in multi-point dynamic task," Int. J. of Automation and Control, Vol.14, No.5-6, pp. 571-592, 2020.
- S. Lu, B. Xin, H. Zhang, and J. Chen, "Agent-based self-organized constructive heuristics for travelling salesman problem," 59th IEEE Conf. on Decision and Control (CDC), pp. 1164-1169, 2020.

Name:
Bin Xin

Affiliation:
Professor, State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology

Address:
No.5 Zhongguancun South Street, Haidian District, Beijing 100081, China

Brief Biographical History:
2012- Lecturer, School of Automation, Beijing Institute of Technology
2015- Associate Professor, School of Automation, Beijing Institute of Technology
2018- Professor, School of Automation, Beijing Institute of Technology

Main Works:
- B. Xin, J. Chen, J. Zhang, H. Fang, and Z.-H. Peng, "Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: A review and taxonomy," IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol.42, No.5, pp. 744-767, 2012.
- Y. Ding, B. Xin, and J. Chen, "Curvature-constrained path elongation with expected length for Dubins vehicle," Automatica, Vol.108, Article No.108495, 2019.

Membership in Academic Societies:
- The Institute of Electrical and Electronics Engineers (IEEE) Computational Intelligence Society
- The Institute of Electrical and Electronics Engineers (IEEE) Control Systems Society
- Chinese Association of Automation (CAA)

Name:
Zhiming Yuan

Affiliation:
Professor, Signal and Communication Research Institute, China Academy of Railway Sciences Co., Ltd.

Address:
No.2 Daliushu Road, Haidian District, Beijing 100081, China

Brief Biographical History:
2019- Professor, Signal and Communication Research Institute, China Academy of Railway Sciences Co., Ltd.

Main Works:
- Z. Yuan, L. Yan, Y. Gao, T. Zhang, and S. Gao, "Virtual parameter learning-based adaptive control for protective automatic train operation," IEEE Trans. on Intelligent Transportation Systems, Vol.22, No.12, pp. 7943-7954, 2021.