A Comparative Study on Evolutionary Algorithms for High-Speed Railway Train Timetable Rescheduling Problem

Shuxin Ding, Tao Zhang, Rongsheng Wang, Chunde Zhang, Sai Lu, and Bin Xin

Signal and Communication Research Institute, Center of National Railway Intelligent Transportation System Engineering and Technology, China Academy of Railway Sciences Corporation Limited
Postgraduate Department, China Academy of Railway Sciences
China Railway Beijing Group Corporation Limited
School of Automation, Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology

The 7th International Workshop on Advanced Computational Intelligence and Intelligent Informatics (IWACIII2021) Beijing, China, Oct.31-Nov.3, 2021
Outline

• Introduction

• Model Formulation

• Evolutionary Algorithms for Train Timetable Rescheduling

• Computational Experiments

• Concluding Remarks
Introduction
A Comparative Study on EAs for High-Speed Railway Train Timetable Rescheduling Problem

China High-Speed Railway (HSR) —— 37900 kilometers

Operation as a network only in China

China High-Speed Railway Network

It is a great challenge to keep the HSR operate punctually

<table>
<thead>
<tr>
<th>Large network size</th>
<th>High operation speed</th>
<th>High traffic density</th>
<th>Large amount of operation</th>
<th>Complex transportation organization</th>
<th>Diversified travel demand</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Country</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>China</td>
<td>CR</td>
</tr>
<tr>
<td>Japan</td>
<td>SCMagle</td>
</tr>
<tr>
<td>France</td>
<td>TGV</td>
</tr>
<tr>
<td>Spain</td>
<td>AVE</td>
</tr>
<tr>
<td>Germany</td>
<td>ICE</td>
</tr>
<tr>
<td>Korea</td>
<td>KTX</td>
</tr>
<tr>
<td>Italy</td>
<td>Frecciarossa 1000</td>
</tr>
<tr>
<td>Italy</td>
<td>Italo</td>
</tr>
<tr>
<td>Turkey</td>
<td>YHT</td>
</tr>
</tbody>
</table>

350km/h

320km/h

300km/h

250km/h
Train Timetable Rescheduling is the key issue for emergency decision under disruption

• If the dispatching is not reasonable, once an emergency occurs, it is easy to cause a large area of train delay and other serious consequences, bringing inconvenience to passengers and reducing the operation efficiency of high-speed railway
How to propose a simple and effective rescheduling model and a fast solution algorithm has become an urgent need for the efficient operation of high-speed railway in China!

Train dispatching system is the "brain" and "commander" of high-speed railway system

**Real Application**
- Mainly handled by dispatchers based on their experience under emergencies

**Theoretical research**
- Formulate mixed integer linear programming models
- Use exact method, metaheuristics, or AI technique

Manual scheduling decision is not optimal decision, which cannot guarantee high efficiency and precise operation

① NP-hard
② Time consuming and suboptimal
Different levels in train scheduling

- **Strategic level**: Network planning, Passenger transport demand analysis, Line planning
- **Tactical level**: Train timetable scheduling, Station operational planning, Rolling stock scheduling, Crew scheduling
- **Operational level**: Train timetable rescheduling, Rolling stock rescheduling, Crew rescheduling

**Branches**:
- Macro demand
- Distributed iterative scheduling
- Artificial experience rescheduling
Paper Contribution

• The high-speed railway train timetable rescheduling problem with a complete station blockage is proposed and modeled as a MILP problem.

• An effective permutation encoding method is proposed for the TTR problem, and a rule-based decoding method is designed to obtain a new schedule. These encoding and decoding methods can manage the entire constraints and guarantee the feasibility of the solution.

• Several evolutionary algorithms are used for solving TTR. Experimental results show that SaDE can efficiently solve most of the test instances compared with other algorithms.
Model Formulation
## Decision Variables

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{i,s}^a )</td>
<td>the actual arrival time of train ( i ) at station ( s )</td>
</tr>
<tr>
<td>( t_{i,s}^d )</td>
<td>the actual departure time of train ( i ) at station ( s )</td>
</tr>
<tr>
<td>( q_{i,j,(s,s+1)} )</td>
<td>the actual traversing order, 1 if train ( i ) traverses on section ( (s, s+1) ) before train ( j ); 0 otherwise</td>
</tr>
<tr>
<td>( y_{i,s} )</td>
<td>the actual train stop indicator, 1 if train ( i ) stops at station ( (s, s+1) ); 0 otherwise</td>
</tr>
</tbody>
</table>

\[
t_{i,s}^a, t_{i,s}^d \geq 0 \quad q_{i,j,(s,s+1)}, y_{i,s} \in \{0, 1\}
\]
Formulation

Objective function

- **Minimize** the total delay time, including the delay arrival and departure time of each train at all the stations

\[
\min \sum_{i \in T} \sum_{s \in S} w_i (t_{i,s}^a - T_{i,s}^a + t_{i,s}^d - T_{i,s}^d)
\]
Formulation

Constraints

• Minimum dwelling time constraints
• Minimum running time constraints
• Headway constraints for departure headway and arrival headway
• Traverse order constraint of two trains in a section
• The arrival and departure times for the unaffected trains are equal to the original timetable
• No trains are allowed to arrive at stations during the disruption
• Timetable constraints that restrict trains are not allowed to arrive and depart from stations before the original arrival and departure time
• The actual traversing orders of all trains are equal to the traversing orders in their first section
• Train stop indicator constraints
Formulation

Constraints

• Minimum dwelling time constraints
• Minimum running time constraints
• Headway constraints for departure headway and arrival headway
• Traverse order constraint of two trains in a section
• The arrival and departure times for the unaffected timetable
• No trains are allowed to arrive at stations during
• Timetable constraints that restrict trains are not at stations before the original arrival and departure
• The actual traversing orders of all trains are equal to first section
• Train stop indicator constraints

\[
\begin{align*}
\text{s.t. } & t^d_{i,s} - t^d_{i,s} \geq d_{i,s} \forall i \in \mathcal{T}; s \in \mathcal{S} \\
& t^a_{i,s+1} - t^d_{i,s} \geq r^{\text{min}}_{i,(s,s+1)} + r^x_{i,(s,s+1)}y_{i,s+1} + r^y_{i,(s,s+1)}y_{i,s+1} \\
& \forall i \in \mathcal{T}; s \in \mathcal{S} \setminus \mathcal{D}(i) \\
& t^d_{j,s} - t^d_{i,s} \geq h(s,s+1)q_{i,j,(s,s+1)} - M(1 - q_{i,j,(s,s+1)}) \\
& \forall i, j \in \mathcal{T}; i \neq j; s \in \mathcal{S} \setminus \mathcal{D}(i) \\
& t^a_{j,s+1} - t^a_{i,s+1} \geq h(s,s+1)q_{i,j,(s,s+1)} - M(1 - q_{i,j,(s,s+1)}) \\
& \forall i, j \in \mathcal{T}; i \neq j; s \in \mathcal{S} \setminus \mathcal{D}(i) \\
& q_{i,j,(s,s+1)} + q_{j,i,(s,s+1)} = 1 \forall i, j \in \mathcal{I}; i \neq j; s \in \mathcal{S} \setminus \mathcal{D}(i) \\
& t^a_{i,s} = T^a_{i,s} \forall i \in \mathcal{T}; s \in \mathcal{S} : T^a_{i,s} \leq H^s_{\text{dis}} \\
& t^d_{i,s} = T^d_{i,s} \forall i \in \mathcal{T}; s \in \mathcal{S} : t^d_{i,s} \leq H^s_{\text{dis}} \\
& t^a_{i,s} \geq H^s_{\text{dis}} + D_{\text{dis}} \forall i \in \mathcal{T} : H^s_{\text{dis}} \leq T^a_{i,s} \leq H^s_{\text{dis}} + D_{\text{dis}} \\
& t^a_{i,O(i)} = t^d_{i,O(i)} \forall i \in \mathcal{T} \\
& t^a_{i,s} \geq T^a_{i,s} \forall i \in \mathcal{T}; s \in \mathcal{S} \\
& t^d_{i,s} \geq T^d_{i,s} \forall i \in \mathcal{T}; s \in \mathcal{S} \\
& q_{i,j,(O(i),O(i) + 1)} = q_{i,j,(s,s+1)} \forall i, j \in \mathcal{T}; i \neq j; s \in \mathcal{S} \setminus \{O(i), D(i)\} \\
& y_{i,s} \leq t^d_{i,s} - t^a_{i,s} \forall i \in \mathcal{T}; s \in \mathcal{S} \setminus \{O(i), D(i)\} \\
& y_{i,s} \geq t^d_{i,s} - t^a_{i,s} \forall i \in \mathcal{T}; s \in \mathcal{S} \setminus \{O(i), D(i)\} \\
& y_{i,s} \geq Y_{i,s} \forall i \in \mathcal{T}; s \in \mathcal{S} \setminus \{O(i), D(i)\} \\
& y_{i,s} = Y_{i,s} \forall i \in \mathcal{T}; s \in \{O(i), D(i)\}
\end{align*}
\]
Formulation

Constraints

- Minimum dwelling time constraints
- Minimum running time constraints
- Headway constraints for departure headway and arrival headway
- Traverse order constraint of two trains in a section
- The arrival and departure times for the unaffected timetable
- No trains are allowed to arrive at stations during
- Timetable constraints that restrict trains are not to arrive at stations before the original arrival and departure
- The actual traversing orders of all trains are equal to the original orders

\[
\text{s.t. } t_{i,s}^d - t_{i,s}^a \geq d_{i,s} \quad \forall i \in T; s \in S
\]
\[
t_{i,s+1}^a - t_{i,s}^a \geq r_{i,s+1}^{\text{min}} + r_{i,s+1}^r y_{i,s} + r_{i,s+1}^e y_{i,s+1} \quad \forall i \in T; s \in S \setminus D(i)
\]
\[
t_{j,s}^d - t_{i,s}^a \geq h_{i,s}(s+1) q_{i,j(s+1)} - M(1 - q_{i,j(s+1)}) \quad \forall i, j \in T; i \neq j; s \in S \setminus D(i)
\]
\[
t_{j,s+1}^a - t_{i,s+1}^a \geq h_{i,s+1} q_{i,j(s+1)} - M(1 - q_{i,j(s+1)}) \quad \forall i, j \in T; i \neq j; s \in S \setminus D(i)
\]
\[
q_{i,j(s+1)} + q_{j,i(s+1)} = 1 \quad \forall i, j \in I; i \neq j; s \in S \setminus D(i)
\]

The problem is an **mixed integer linear programming** problem which belongs to **NP-hard**.
Evolutionary Algorithms for Train Timetable Rescheduling
Encoding and Decoding

- Using permutation-based encoding instead of real-coded encoding
- Real-coded encoding
  \[ [t_{1,1}, t_{1,2}, t_{2,1}, t_{2,2}, \ldots, t_{i,s}, t_{i,s}, \ldots, t_{|T||S|}, t_{|T||S|}], i \in T, s \in S, 1 \leq t_{i,s}, t_{i,s} \leq 1440 \]
  - Dimension: 2|T||S|
  - Solution space: 1440^2|T||S| (for integer arrival/departure time)
- Permutation-based encoding
  \[ [p_1, p_2, \ldots, p_i, \ldots, p_{|T|}], i \in T, p_i \in \{1, \ldots, |T|\}, p_i \neq p_j : i \neq j \]
  - Dimension: |T|
  - Solution space: |T|!
- The dimension and solution space is much smaller in permutation-based encoding
- There are unfeasible region in real-coded encoding, constraints handling should be designed
Encoding and Decoding

- Obtain the actual arrival time and departure time through the decoding procedure
  - Traversing order is obtained through the permutation-based encoding
  - Decide arrival time and departure time satisfying different constraints

A Comparative Study on EAs for High-Speed Railway Train Timetable Rescheduling Problem

Minimum running time constraints

Minimum dwelling time constraints
Encoding and Decoding

- Obtain the actual arrival time and departure time through the decoding procedure
  - Traversing order is obtained through the permutation-based encoding
  - Decide arrival time and departure time satisfying different constraints

Minimum running time constraints

Minimum dwelling time constraints
Encoding and Decoding

• Obtain the actual arrival time and departure time through the decoding procedure
  • Traversing order is obtained through the permutation-based encoding
  • Decide arrival time and departure time satisfying different constraints

Headway constraints

Station \( j \)
Section \( k \)
Station \( j+1 \)
Minimal headway

Station \( j \)
train \( i \)
Station blockage

Depart after disruption ends
Encoding and Decoding

• Obtain the actual arrival time and departure time through the decoding procedure
  • Traversing order is obtained through the permutation-based encoding
  • Decide arrival time and departure time satisfying different constraints

A Comparative Study on EAs for High-Speed Railway Train Timetable Rescheduling Problem
Encoding and Decoding

Algorithm 1 Decoding Procedure

Input: The original timetable information; The disruption information; The set of affected trains $T_{	ext{dis}}$; Scheduling order of the trains $p = [p_1, \ldots, p_l]$.

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

1. for $i = 1$ to $|T| - |T_{	ext{dis}}|$ 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 $i = |T| - |T_{	ext{dis}}| + 1$ to $|T|$ do
7. if $i = |T| - |T_{	ext{dis}}| + 1$ then
8. $t^a_{p_i, O(p_i)} = H^d_{i,s} + D^d_{i,s}$;
9. $t^d_{p_i, O(p_i)} = t^a_{p_i, O(p_i)}$;
10. else
11. $t^a_{p_i, O(p_i)} = \max(t^a_{p_{i-1}, O(p_{i-1})} + h(\delta(p_i), O(p_i) + 1), T^a_{p_i, O(p_i)})$;
12. $t^d_{p_i, O(p_i)} = \max(t^a_{p_i, O(p_i)} + d_{p_i, O(p_i)}), T^d_{p_i, O(p_i)})$;
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. $t^a_{p_i, s} = \max(t^a_{p_i, s-1} + r_{\text{min}}_{p_i, \delta(s-1, s)} + y_{p_i, s-1} r_{\text{p}_i, \delta(s-1, s)} + y_{p_i, s} r_{\text{p}_i, \delta(s-1, s)} - T^a_{p_i, s}, T^a_{p_i, s})$;
18. $t^d_{p_i, s} = \max(t^d_{p_i, s}, T^d_{p_i, s})$;
19. $y_{p_i, s} = \max(t^d_{p_i, s} + d_{p_i, s}, T^d_{p_i, s})$;
20. if $s < D(p_i)$ then
21. $t^d_{p_i, s} = \max(t^d_{p_i, s}, t^d_{p_i, s} + h(s, s+1))$;
22. if $\text{sgn}(t^d_{p_i, s} - t^a_{p_i, s}) > y_{p_i, s}$ then
23. $t^a_{p_i, s} = \min(t^d_{p_i, s}, t^a_{p_i, s} + r_{\text{min}}_{p_i, \delta(s-1, s)} + y_{p_i, s-1} r_{\text{p}_i, \delta(s-1, s)} + r_{\text{p}_i, \delta(s-1, s)} - T^a_{p_i, s}, T^a_{p_i, s})$;
24. $y_{p_i, s} = \text{sgn}(t^d_{p_i, s} - t^a_{p_i, s})$;
25. end if
26. end if
27. end for
28. end for
29. return
Evolutionary Algorithms

• A Dual-Model Estimation of Distribution Algorithm (DM-EDA)
• Self-adaptive Differential Evolution (SaDE)
• Comprehensive Learning Particle Swarm Optimizer (CLPSO)
• Covariance Matrix Adaptation Evolution Strategy (CMA-ES)
Evolutionary Algorithms

• A Dual-Model Estimation of Distribution Algorithm (DM-EDA)
  • It estimates the overall distribution of the parent solutions and updates a probabilistic model with the superior individuals
  • New solutions are sampled from the model
  • Node histogram model (NHM) and edge histogram model (EHM) are selected for permutation-based optimization problem
  • Truncation selection and restart strategy are used

DM-EDA is designed to search in discrete space
Evolutionary Algorithms

• Self-adaptive Differential Evolution (SaDE)
  • It uses a self-adaptive method to choose trial vector generation strategies and control parameter values

• Comprehensive Learning Particle Swarm Optimizer (CLPSO)
  • Each dimension of a particle learns from the best corresponding dimension of the particle

• Covariance Matrix Adaptation Evolution Strategy (CMA-ES)
  • It also uses a probability model to obtain new solutions
  • It samples solutions from a multivariate normal distribution

All above algorithms are designed to search in continuous space
Evolutionary Algorithms

• A Dual-Model Estimation of Distribution Algorithm (DM-EDA)
• Self-adaptive Differential Evolution (SaDE)
• Comprehensive Learning Particle Swarm Optimizer (CLPSO)
• Covariance Matrix Adaptation Evolution Strategy (CMA-ES)
• **Random Key Algorithm** for algorithms designed to search in continuous space (SaDE, CLPSO, and CMA-ES)

real vector (3.5, 2.4, 1.6, 0.5, 4.1) → permutation (4, 3, 2, 1, 5)
Computational Experiments
Computational Experiments

- The Beijing–Tianjin intercity HSR timetable
- 6 stations and 5 sections
- 40 trains downstream from 6:00 to 12:00
- Dwell time: 2 min
- Minimum running time of each section are 5, 5, 6, 5, 5 (min), respectively
- Additional times caused by starting and stopping are 2 min and 3 min
- Minimal headway: 4 min

Fig. 1. Original timetable for Beijing–Tianjin intercity railway with 40 downstream trains within 6-h time horizon.
Computational Experiments

- 8 test instances from 2 cases on train weights
  - Case 1: The weight values of trains are set to 1.
  - Case 2: The weight values of trains are generated as uniformly distributed random integers in a range between 1 to 10.

- $|T|$ is the total trains considered
- $D_{dis}$ is the disruption duration

| Table 3. Setting of the two basic parameters for the test instances. |
|---|---|---|---|---|
| No. | $|T|$ | $D_{dis}$ (min) | No. | $|T|$ | $D_{dis}$ (min) |
| 1, 5 | 15 | 30 | 2, 6 | 20 | 50 |
| 3, 7 | 30 | 70 | 4, 8 | 40 | 90 |
Computational Experiments

Table 4. Results of the comparison between DM-EDA, SaDE, CLPSO, and CMA-ES.

<table>
<thead>
<tr>
<th>No.</th>
<th>DM-EDA</th>
<th>SaDE</th>
<th>CLPSO</th>
<th>CMA-ES</th>
<th>CPLEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1628.0000 ± 0.0000</td>
<td>1628.0000 ± 0.0000</td>
<td>1628.0000 ± 0.0000</td>
<td>1628.0000 ± 0.0000</td>
<td>1628.0000</td>
</tr>
<tr>
<td>2</td>
<td>3874.0000 ± 0.0000</td>
<td>3874.0000 ± 0.0000</td>
<td>3874.0000 ± 0.0000</td>
<td>3874.0000 ± 0.0000</td>
<td>3874.0000</td>
</tr>
<tr>
<td>3</td>
<td>7570.8000 ± 34.5522</td>
<td>7272.8000 ± 7.5226</td>
<td>7274.4000 ± 7.6116</td>
<td>7284.3000 ± 0.7327</td>
<td>7268.0000</td>
</tr>
<tr>
<td>4</td>
<td>12539.2000 ± 55.0154</td>
<td>12070.0000 ± 0.0000</td>
<td>12072.1000 ± 3.3388</td>
<td>12081.7000 ± 13.2709</td>
<td>12070.0000</td>
</tr>
<tr>
<td>5</td>
<td>6462.0000 ± 0.0000</td>
<td>6126.0000 ± 0.0000</td>
<td>6126.0000 ± 0.0000</td>
<td>6126.0000 ± 0.0000</td>
<td>6126.0000</td>
</tr>
<tr>
<td>6</td>
<td>15386.0000 ± 0.0000</td>
<td>14810.0000 ± 0.0000</td>
<td>14810.0000 ± 0.0000</td>
<td>15060.6000 ± 695.6067</td>
<td>14810.0000</td>
</tr>
<tr>
<td>7</td>
<td>31475.0500 ± 684.5033</td>
<td>26874.6000 ± 8.0026</td>
<td>26875.3000 ± 8.3168</td>
<td>27177.0000 ± 330.6453</td>
<td>26872.0000</td>
</tr>
<tr>
<td>8</td>
<td>59492.1000 ± 1055.7585</td>
<td>43125.0000 ± 10.7508</td>
<td>43636.0000 ± 157.0169</td>
<td>43697.0000 ± 599.0109</td>
<td>43128.0000</td>
</tr>
</tbody>
</table>

1. CPLEX stopped after running for one hour.
2. Optimal value.

- In five instances (No. 1, 2, 4, 5, and 6), the results of SaDE equal that of CPLEX.
- Moreover, for instances No. 3 and 7, the results of SaDE are only slightly larger (0.07% and 0.01%) than that of CPLEX (within one hour).
- In instance No. 8, the result of SaDE is better than that of CPLEX (within one hour).
Computational Experiments

• Converge curves of the four EAs in instances No. 3, 4, 7, and 8
• The curves are zoomed in some areas for better visualization
• CMA-ES converges faster than other algorithms
• SaDE converges second but provides better results.

Fig. 2. Convergence curves of the proposed DM-EDA, SaDE, CLPSO, and CMA-ES for several test instances.
Computational Experiments

Table 5. Runtime performance of different algorithms (sec.).

<table>
<thead>
<tr>
<th>No.</th>
<th>DM-EDA</th>
<th>SaDE</th>
<th>CLPSO</th>
<th>CMA-ES</th>
<th>CPLEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.7372 ± 0.4578</td>
<td>8.1800 ± 0.6489</td>
<td>3.7526 ± 0.2992</td>
<td>2.2386 ± 0.3614</td>
<td>10.3855</td>
</tr>
<tr>
<td>2</td>
<td>10.0875 ± 0.6872</td>
<td>11.9927 ± 0.6156</td>
<td>5.5700 ± 0.3859</td>
<td>3.0539 ± 0.2743</td>
<td>64.7492</td>
</tr>
<tr>
<td>3</td>
<td>24.8920 ± 0.9677</td>
<td>19.9135 ± 1.2381</td>
<td>11.2691 ± 1.8045</td>
<td>6.0701 ± 1.0042</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>47.5454 ± 1.8224</td>
<td>30.0148 ± 2.1255</td>
<td>17.3618 ± 0.6499</td>
<td>9.6739 ± 0.1859</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>5.1246 ± 0.5452</td>
<td>8.1143 ± 1.1712</td>
<td>3.7058 ± 0.6896</td>
<td>1.8761 ± 0.3053</td>
<td>10.5488</td>
</tr>
<tr>
<td>6</td>
<td>10.2779 ± 1.2930</td>
<td>12.3090 ± 2.0349</td>
<td>6.0593 ± 0.7121</td>
<td>2.7641 ± 0.1174</td>
<td>30.5911</td>
</tr>
<tr>
<td>7</td>
<td>24.8921 ± 0.8240</td>
<td>20.0972 ± 1.1450</td>
<td>11.3079 ± 1.3316</td>
<td>6.2739 ± 1.1048</td>
<td>2861.8612</td>
</tr>
<tr>
<td>8</td>
<td>49.8737 ± 3.1044</td>
<td>31.1891 ± 2.9282</td>
<td>17.4095 ± 0.8151</td>
<td>10.4551 ± 1.6275</td>
<td>-</td>
</tr>
</tbody>
</table>

* CPLEX cannot find optimal value after running for one hour.

• The result shows that DM-EDA takes the longest time compared with the other EAs.
• All instances can be solved within one minute.
• The running time for CPLEX increases. For some instances, it is more than 1 hour.
Concluding Remarks

• The high-speed railway TTR problem is formulated as a MILP problem.
• Four EAs are designed to solve TTR.
• A novel encoding and decoding method are specially designed.
• Obtained optimal/suboptimal solutions within one minute.

Future Research

• Consider situations with more types of trains (e.g., trains with different prefixes including G, C, D).
• Consider reordering in other stations.
• Consider the uncertainties in the dynamic environment.
Thank you for your attention!

Q&A
Motivation

• High-speed railway (HSR) may face inevitable emergencies, e.g., infrastructure failure, train failure, natural disasters.

• When the scale of the problem is getting larger, using the CPLEX solver will cost much time, which may exceed the time limit.

• Unlike past works use real-encoding based metaheuristics.

We introduce a novel train timetable rescheduling problem with a complete station blockage as an mixed integer linear programming and considers an effective permutation-based metaheuristics to solve the problem with near-optimal/optimal solutions in real-time.