Multi-Objective Optimization for Train Timetable Rescheduling with Appropriate Robustness Against Uncertain Arrival Deviations

Yukun Wang,Xuewu Dai, *Member, IEEE*,Yixiang Yue,Dongliang Cui,Guoqi Feng,Shengping Yu, *Member, IEEE*

***Abstract*—Train operations are vulnerable to perturbations with significant uncertainty, resulting in deviations between actual and scheduled arrival times of varying severity. Without sufficient robustness to absorb potential future arrival deviations that are highly likely to occur, a rescheduled timetable tends to become ineffective and may demand frequent adjustments. Determining the appropriate extent to which the robustness of a rescheduled timetable should be enhanced is challenging, as such improvements introduce certain negative impacts on railway systems, and the acceptable level of robustness differs across different rescheduling scenarios. To balance the conflict between timetable robustness and its negative impacts, this paper proposes a multi-objective optimization model. The statistical distribution of arrival deviations is employed to capture the underlying uncertainty. By adding interval determining variables into the constraints to improve the robust optimization model, our model makes rescheduled timetables able to resist frequent arrival deviations within a predefined interval. For generating alternative rescheduled timetables with varying levels of robustness, the upper and lower bounds of the predefined uncertainty interval are treated as decision variables. To reasonably reduce the number of adjustments, our model formulates delay propagation risk and total delays as objective functions. A multi-population non dominated sorting genetic algorithm II (MP-NSGA-II) is designed to efficiently approximate the model's Pareto-optimal set (POS). Finally, the method's effectiveness is validated based on the Elizabeth Line. The results demonstrate that our method can provide effective decision support for dispatchers in improving the robustness of rescheduled timetables against perturbations.**

***Index Terms*—Multi-objective optimization, evolutionary algorithm, robust train timetable rescheduling, uncertain perturbation, stochastic programming, robust optimization .**

I. Introduction

As a reliable, efficient, and convenient mode of transportation, railway systems have expanded rapidly in various countries in recent decades [1]. Trains, passengers, staff, and other elements within the systems are efficiently organized through meticulously planned train timetables. However, railway systems are vulnerable to uncertain perturbations such as human error, equipment failures and bad weather, which often lead to the actual arrival time of trains deviating from the scheduled times in the official timetable [2]. When significant delays occur, it is essential to promptly reschedule the train timetable; otherwise, delays will quickly propagate among trains and stations, seriously reducing passenger satisfaction. Thus, train timetable rescheduling (TTR) is a vital measure to maintain the smooth operation of railway systems.

To ensure the actual application effectiveness of rescheduled timetables in complex environments, rescheduled timetables should possess a certain degree of robustness against potential future arrival deviations. Uncertain perturbations, such as strong wind, exist continuously and affect railway systems, making the section running status of the train highly uncertain. Arrival deviations may occur at any time after a rescheduled timetable is established, impacting its practical application. Without this robustness, if a train arrives earlier than the rescheduled time, it may exceed speed limits, compromising safety. Conversely, if a train arrives later, even minor delays can render the rescheduled timetable ineffective. Therefore, rescheduled timetables should be robust against potential future arrival deviations. Most existing studies focus on generating feasible solutions within a short computation time, while tending to overlook these potential arrival deviations [3]. As a result, although these approaches consider complex factors such as passengers, multi-line interactions, and diverse rescheduling strategies, their solutions may quickly become infeasible when affected by arrival deviations [4], [5].

A flexible method that enables rescheduled timetables to have different degrees of robustness, ensuring their continued applicability under the impact of those arrival deviations that are likely to occur in the future, can help dispatchers generate robust rescheduled timetables that better meet the practical needs of different scenarios, which is of great significance to railway operation management. Improving timetable robustness is challenging, as making a timetable resilient to more significant deviations requires allocating more time supplements on top of the technically minimal operational process times [6]. However, this leads to more train delays and decreases the overall operational efficiency of the system. The busier the passenger transport mission is, where maintaining efficient operations is crucial, the more unacceptable these time

Yukun Wang, Dongliang Cui and Shengping Yu are with the State Key Labo ratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China (e-mail: 2210373@stu.neu.edu.cn; cuidongliang@mail.neu.edu.cn;spyu@mail.neu.edu.cn).

Xuewu Dai is with the Department of Mathematics, Physics and Electrical Engineering, Northumbria University, Newcastle upon Tyne NEl 8ST, U.K. (e-mail: xuewu.dai@northumbria.ac.uk).

Yixiang Yue is with the Department of Frontiers Science Center for Smart High-Speed Railway System, School of Traffic and Transportation,Beijing Jiaotong University, Beijing 100044, China (e-mail: yxyue@bjtu.edu.cn).

Guoqi Feng is with the School of Business Administration, Northeastern University, Shenyang 110169, China (e-mail: gqfeng@mail.neu.edu.cn).

0000–0000/00$00.00 © 2021 IEEE

supplements become. Although several methods have been proposed to enhance the robustness of rescheduled timetables [7]–[9], they tend to overlook a significant characteristic of the statistical distribution of arrival deviations: minor deviations occur more frequently, while large deviations are relatively rare. Therefore, rescheduled timetables should be designed to be robust against deviations that occur with higher frequency. Moreover, the level of robustness should be flexibly adjustable and controllable, allowing the generation of robust rescheduled timetables that align with the acceptable degree of robustness required under specific operational conditions.

When improving the robustness of rescheduled timetables against potential future arrival deviations, we are very concerned about whether the number of adjustments to rescheduled timetables required in the future can be effectively reduced, as this will greatly facilitate the operational management of railway systems. Because these adjustments often require human participation, this metric is closely related to the workload of dispatchers. Furthermore, frequent adjustments lead dispatchers to frequently broadcast and coordinate multiple versions of the timetable, which adds an extra burden to the management of the railway system [10].

Since improving the robustness of rescheduled timetables against potential future arrival deviations will also lead to certain negative impacts, a multi-objective optimization approach that can balance these conflicting objectives is needed to help dispatchers generate a rescheduled timetable with appropriate robustness. Particularly in situations where the tolerance for such negative impacts is extremely low, it is essential to carefully evaluate whether further enhancing robustness is truly appropriate. If the number of adjustments can be significantly reduced at a small cost within an acceptable range, then such a rescheduled timetable with appropriate robustness is highly desirable. However, when robustness has been improved to a certain level, further enhancing robustness will not effectively reduce the number of adjustments but will instead bring too many negative effects. In this case, a solution with excessive robustness becomes unreasonable.

It is worth noting that a major feature of TTR is that it has a strong stringent real-time requirement, unlike other scheduling problems (e.g., train timetable optimization). Dispatchers are typically required to regenerate a TTR plan in only a few minutes [11]. This requires that any decision support method designed for TTR not only has the ability to respond quickly. It also needs to be able to stably and efficiently generate a diverse set of high-quality rescheduled timetables within a short time frame. Importantly, these candidate solutions should exhibit varying levels of robustness, thereby enabling dispatchers to select from a range of options according to the current operational context.

Motivated by the aforementioned challenges, this paper focuses on how to provide effective decision support for dispatchers when the train operation status is affected by perturbations with certain uncertainty, which leads to repeated failures of the rescheduled timetables, so as to reasonably reduce the number of adjustments to the rescheduled timetables. This paper contributes to TTR literature as follows:

1) A novel mathematical constraint formulation is constructed to enable rescheduled timetables to resist potential future arrival deviations that are most likely to occur. First, the statistical distribution of arrival deviation is used to capture its uncertainty, and this random variable is incorporated into the constraints. Second, the proposed constraints integrate stochastic programming with robust optimization, and incorporate interval-determining variables of random variables, thereby ensuring that the solution remains feasible even affected by arrival deviations within a defined interval.
2) A method to make the rescheduled timetables have adjustable robustness against potential future arrival deviations is proposed. The upper and lower bounds of the interval of arrival deviations are modeled as a new type of decision variables, enabling the model to flexibly generate rescheduled timetables with tailored robustness levels in response to different rescheduling scenarios.
3) A multi-objective optimization approach that can reasonably reduce the number of adjustments to rescheduled timetables is designed. Firstly, a mathematical model with two objective functions, specifically delay propagation risk and total delays, is developed to indirectly measure the number of adjustments to the robust rescheduled timetable and its negative impact on the rail transit system. Additionally, by generating the Pareto-optimal set and corresponding Pareto front of the model, a range of alternative robust rescheduled timetables is offered, assisting dispatchers in better balancing the two objective functions.
4) A multi-population non-dominated sorting genetic algorithm II (MP-NSGA-II) is developed to efficiently solve the proposed model. To enhance the performance of NSGA-II, a variety of strategies are combined to comprehensively improve it. First, the entire search space is partitioned into several distinct subregions, with multiple populations collaboratively exploring through a division-of-labor mechanism. Furthermore, the local search capability is strengthened by combining external archiving with neighborhood search. During each iteration, elite individuals are fed back to all sub-populations, effectively guiding the subsequent evolutionary process and accelerating convergence.

The rest of this paper is organized as follows. Section II gives a relevant literature review. Section III introduces the considered problem. Then, the considered problem is formulated as the multi-objective optimization model in Section IV. Section V introduces the algorithm specifically designed to efficiently solve the model. In Section VI, the effectiveness of the proposed method is evaluated based on an actual data set. Finally, the conclusion and future work are discussed in Section VII.

II. Literature review

A. Train timetable rescheduling

Railway systems are highly complex and constantly changing over time. After a rescheduled timetable is established, various unforeseen situations may still arise in the future,

resulting in considerable operational uncertainty. From our perspective, TTR research can be categorized into deterministic rescheduling and robust rescheduling, depending on whether the potential future uncertainties in railway systems are taken into account.

Deterministic rescheduling overlooks potential future uncertainties. It relies on the assumption that all acquired information is accurate and that rescheduled timetables can be executed precisely as intended.

As an important means to restore the normal operation of railway systems, deterministic rescheduling initially was carried out from the perspective of the operator, with total train delays employed as an early objective function [12]. Subsequently, additional metrics related to system operations, such as energy consumption [13] and operating cost [14], have also been developed. Additionally, some studies adopt the passenger perspective, focusing on the impact on passengers. From a passenger-centric view, deterministic rescheduling tries to minimize inconvenience to passengers [15].

Based on the optimization objectives from the operator and passenger perspectives, researchers have explored various amendments strategies in the rescheduling [14], [16], [17]. Researchers investigated rescheduling problems across different line structures [18]–[20]. Additionally, advanced solving methods such as heuristic [21]–[23], meta-heuristic [24], [25], and artificial intelligence techniques [11], [26] have been introduced to the solving process.

Robust rescheduling recognizes that the actual situations in the future are difficult to determine at the moment when rescheduling, and each situation may occur with a certain probability. Robust rescheduling attempts to make the rescheduled timetables adapt to such uncertain environments.

The unexpected perturbations triggering rescheduling inherently possess some uncertainty, and the relevant information used to construct models may change dynamically over time. The uncertainty of disruption duration time is one of the earliest factors to attract researchers' attention and has been widely studied. Considering the uncertainty in recovery times for single-track railway disruptions and section running times, [10] proposed a method that combines stochastic programming with a rolling horizon decision process. The application scenarios of this rolling horizon approach have been further expanded: [27] applied it to double-track railways rescheduling problem in case of a partial segment blockage. Contrary to previous scholars, [28] noted the potential lack of sample data regarding disruption duration and formulated the problem as a credibilistic two-stage fuzzy 0–1 integer optimization model.

Train dwell times and section running times also exhibit significant uncertainty, impacting the practical implementation of rescheduled timetables. To enhance the robustness of rescheduled timetables against disturbances, [7] proposed a Mixed Integer Linear Programming model, which compromise the dual objectives of minimizing delays and maximizing the robustness through a weighted sum approach. [9] formulated this problem as scenario-based chance-constrained model to improve the robustness of the rescheduled timetables against disturbances. [8] also adopted the method of improving the constraints in the mathematical model to enhance the robustness and presented a mixed integer quadratically constrained programming model for the robust train regulation problem. [29] modeled the uncertainty as scenario-based chance constraints to strengthen the robustness of the rescheduling scheme in case of uncertain perturbations.

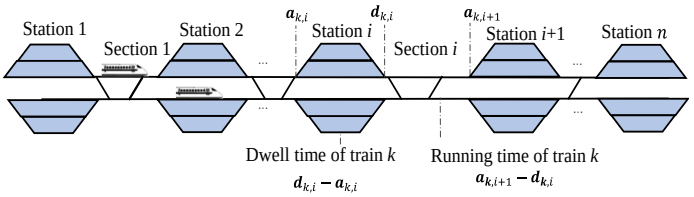


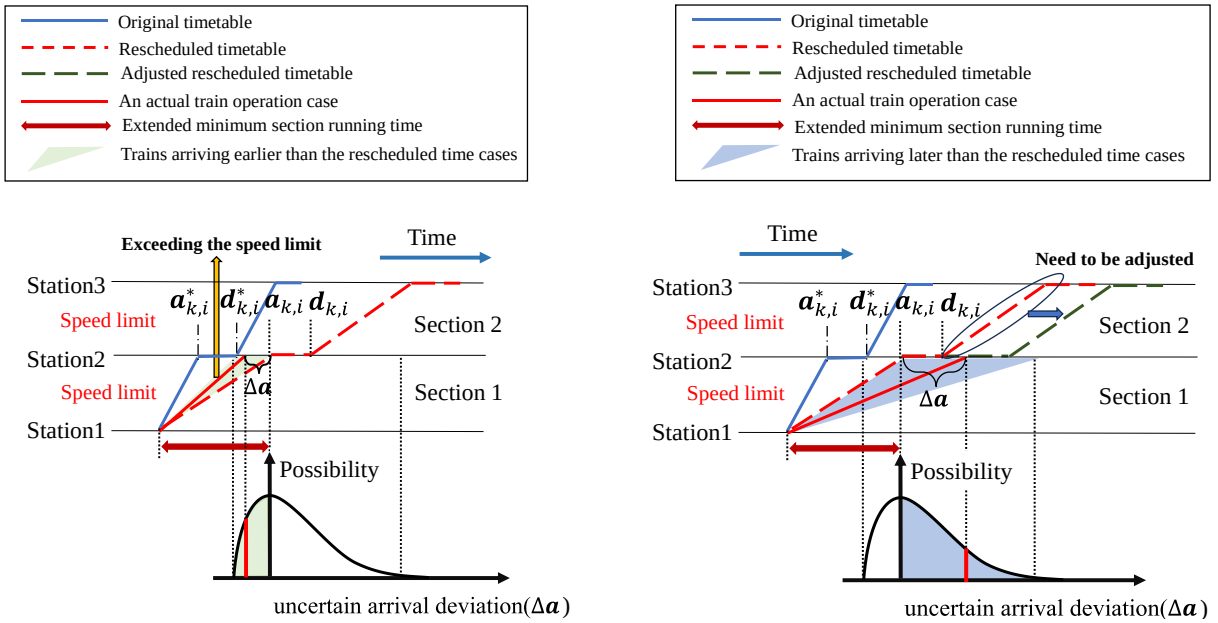
Fig. 1. Illustration of double-track railways.

In existing research on robust rescheduling, an important statistical distribution characteristic of arrival deviations is overlooked: minor deviations occur with high frequency, whereas larger deviations are increasingly rare. While enabling rescheduled timetables to resist larger deviations enhances robustness, it also imposes a heavier operational burden. Therefore, it is essential to carefully consider how to improve the robustness of rescheduled timetables in a more rational and effective manner. Currently, there is a lack of a method to make timetables robust against arrival deviations that are most likely to occur in the future. In addition, there is an absence of a multi-objective optimization approach to assist dispatchers in evaluating to what extent robustness should be enhanced to achieve a desirable trade-off.

B. Applications of NSGA-II to transportation

The non-dominated sorting genetic algorithm II (NSGA-II) is a well-established approach for multi-objective optimization proposed by Deb et al. [30]. In the past 20 years, NSGA-II has been used in many fields because of its excellent performance, such as well diversity, well convergence and fast non-dominated rank [31]. Transportation system is one of the most important application areas [32]. NSGA-II is a preferred choice for addressing multi-objective scheduling problems at container terminals, due to its simplicity, cost efficiency, and significant relevance in comparative research [33]. In the application to multi-objective scheduling problems at container terminals, NSGA-II is continuously improved to enhance its local exploitation capability [34]–[36]. It is a very important problem to cut down operation cost and guarantee service quality for bus vehicle scheduling, and [37] used NSGA-II to choose a subset of vehicle blocks from candidate block set under uncertain environments, to generate a non-dominated solution set.

NSGA-II is worth trying to solve the multi-objective robust rescheduling problem proposed in this paper. The application of NSGA-II faces certain challenges. While NSGA-II excels in global exploration, it exhibits relatively limited local exploitation capabilities. This drawback restricts its capacity to approximate the true Pareto front effectively within limited computational time.



(a) The train arrives earlier than the rescheduled time

(b) The train arrives later than the rescheduled time

Fig. 2. Illustration of uncertain arrival deviation Δa.

III. Problem Description

Section III-A first introduces the stochastic arrival deviations considered in this study. Section III-B provides a detailed elaboration of the more desirable robust rescheduled timetable for this situation, as shown in Fig.4.

A. Uncertain arrival deviations in rescheduling

This paper studies the problem that when trains running on double-track railways (as illustrated in Fig.1) are affected by unexpected perturbations such as strong winds, the train speed is restricted, the original train timetable can no longer be used, and a new timetable needs to be formulated in time. However, there may be a certain deviation Δa between the actual arrival time of trains affected by the uncertainty of unexpected perturbations and the rescheduled times, which is shown in Fig.2. For example, when trains are affected by speed restrictions due to strong winds, the trains' running time in the speed limit section is extended. Since the frequently changing wind speed makes the additional running time in the affected section also changeable, it is difficult to accurately estimate how much the train's section running time will be extended. Trains may arrive either earlier or later than the rescheduled times, which causes unsafe operations or renders the rescheduled timetable invalid. Particularly under the impact of train delays, rescheduled timetables that fail will require further adjustments, which not only increases the workload for staff but also complicates the management of the railway system. Therefore, we hope that the rescheduling plan for trains within the spatiotemporal scope affected by uncertain perturbations can be robust against such arrival deviations, thereby reducing the number of necessary adjustments towards failure-rescheduled timetables.

B. Reasonable robust rescheduled timetables considering uncertain arrival deviations

Reducing the number of adjustments requires adding time supplements to improve the robustness of the rescheduled timetable against uncertain arrival deviations. However, these time supplements will also introduce certain negative impacts on the railway system. If we design a rescheduled timetable that accounts for all possible arrival deviations to avoid any adjustments (shown in Fig.3), it would require a substantial amount of time supplements, significantly reducing operational efficiency, and the probability of long delays occurring is relatively low. Therefore, a robust timetable illustrated in Fig.4, which can withstand frequent arrival deviations within a specific interval, is preferable. To be more specific, even if influenced by $\forall \Delta a \in [\underline{\alpha}, \overline{\alpha}]$, such a robust rescheduled timetable can still ensure that the actual operation of the train meets various specific requirements, namely constraints. Delays outside the interval $[\underline{\alpha}, \overline{\alpha}]$ are still possible, and in such cases, the rescheduled timetable would still need to be adjusted, which is illustrated in Fig.5.

The tolerance for the negative impacts of time supplements and the demand for robustness may vary depending on the specific rescheduling scenario. As a result, the upper and lower bounds of the interval that need to be set may also vary in different scenarios.

In summary, the main challenge in reducing the number of adjustments to the rescheduled timetables is that reducing this indicator will bring certain negative impacts to the railway system, requiring a careful balance between the two. Additionally, the degree of acceptability of such negative impacts may vary in different scenarios. Therefore, it is crucial to propose a method that can help us quantitatively analyze the relationship

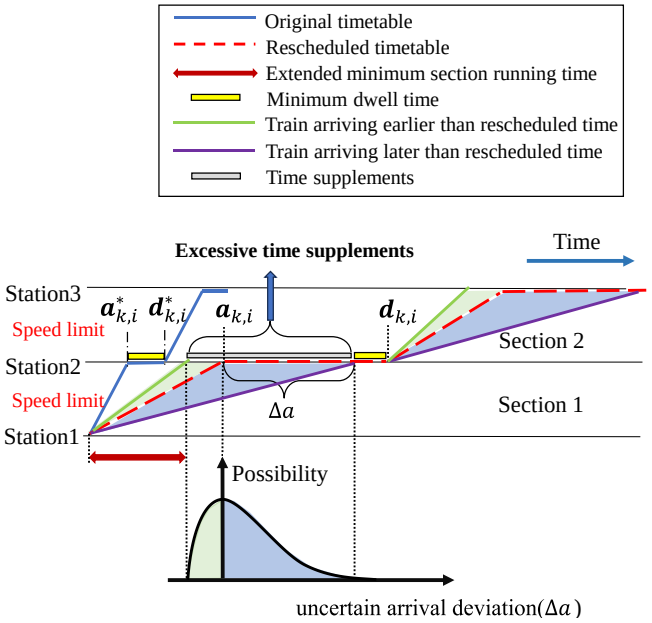


Fig. 3. A rescheduled timetable considering all possible arrival deviations without any further adjustments.

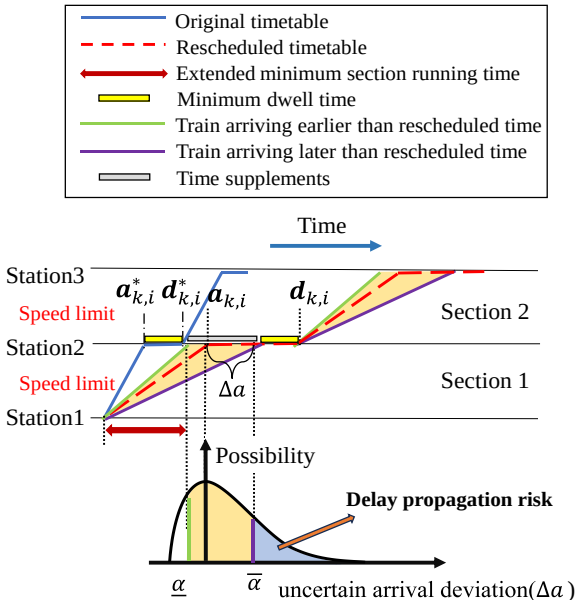


Fig. 4. A more desirable rescheduled timetable that can withstand most arrival deviations.

between the two when improving rescheduled timetables' robustness so as to aid dispatchers in flexibly formulating the most appropriate timetable for specific scenarios.

IV. Mathematical Formulation

To address the proposed robust rescheduling problem, a multi-objective optimization model is formulated in this section, and section IV-A first introduces the notations used in the mathematical formulation. Then, sections IV-B and IV-C respectively introduce the constraints and the objective functions required to properly model this problem in detail.

TABLE I
The Definition of Parameters

Parameters	Description
S	Set of stations in the dispatch region
T	Set of trains to be rescheduled
m	Number of trains
n	Number of stations
i	Subscript for i-th station in set S
PT_i	Set of trains that pass station i
ST_i	Set of trains that stop and dwell at station i
C_i	Set of tracks in station i
$a^*_{k,i}$	Train k's arrival time at station i in the original timetable
$d^*_{k,i}$	Train k's departure time at station i in the original timetable
$x^*_{k,i}$	A binary parameter representing if train k dwells at station i in the original timetable. $x^*_{k,i}$=1 means that train k needs to dwell at station i, otherwise $x^*_{k,i}=0$.
r_i	The minimum running time of section i from station i to station i +1.
r^{ex}_i	Extended minimum running time of section i between station i and station (i +1) due to temporary speed restriction.
H_{min}	The minimum headway for two consecutive trains departing from or arriving at the same station i.
h_{min}	The minimum headway for two consecutive trains using the same track in station i.
$y^*_{g,k,i}$	A binary parameter denoting the running order of two trains g and k at station i in the original timetable. $y^*_{g,k,i}$=1 if tain g travels before train k. Otherwise, $y^*_{g,k,i}=0$.
$z^*_{k,p,i}$	A parameter for track assignment at station i. $z^*_{k,p,i}$=1, if train k uses track p at station i. Otherwise, $z^*_{k,p,i}=0$.
M	A sufficiently large positive value

TABLE II
The Definition of Variables

Random variable	
Δa	Deviation of train k's actual arrival time at station i from the arrival time planned in the rescheduled timetable
Decision variable	
$a_{k,i}$	Arrival time to be rescheduled for train k arriving at station i
$d_{k,i}$	Departure time to be rescheduled for train k departing from station i
$\underline{\alpha}$	An integer variable deciding the minimum value of the arrival deviation Δa that the robust rescheduled timetable needs to resist
$\overline{\alpha}$	An integer variable deciding the maximum value of the arrival deviation Δa that the robust rescheduled timetable needs to resist
Interval-determining variables of random variable	
$\gamma^j_{\underline{\alpha},\overline{\alpha}}$	A binary variable whose value is up to the interval $[\underline{\alpha},\overline{\alpha}]$, is used to determine whether the negative impact of $\forall \Delta a \in [\underline{\alpha},\overline{\alpha}]$ on the rescheduled timetables should be considered in the constraints. $\gamma^j_{\underline{\alpha},\overline{\alpha}}$=1 denotes that $[\underline{\alpha},\overline{\alpha}]$ includes situations that would cause the actual operation of the train to potentially fail to meet the minimum operational requirements, necessitating the consideration of the impact of $\forall \Delta a \in [\underline{\alpha},\overline{\alpha}]$ on the constraints. Otherwise, $\gamma^j_{\underline{\alpha},\overline{\alpha}}$=0. j=1 or 2.

A. Notation Description

For the convenience of modeling, the relevant unexplained parameters and variables are summarized in TABLE I and TA-

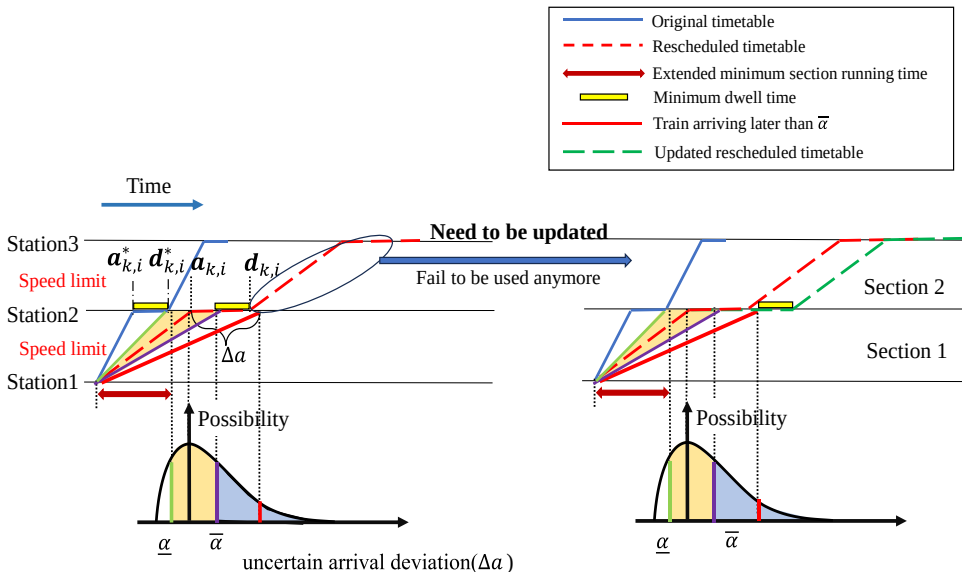


Fig. 5. The ideal robust rescheduled timetable with a certain risk of requiring further adjustments.

BLE II, respectively. Unless otherwise specified, the subscript i denotes a station, and the subscripts k and g represent a train.

B. Constraints

Given the potential future deviations Δa that may exist between the rescheduled arrival time and the actual arrival time of the train, affected by the uncertainty, such deviations may lead to undesirable consequences. Therefore, we aim to incorporate an acceptable level of robustness into the rescheduled timetable, so that the rescheduled timetable can resisting frequent arrival deviations within a specific interval $[\underline{\alpha}, \overline{\alpha}]$ as shown in Fig.4. More specifically, even influenced by $\forall \Delta a \in [\underline{\alpha}, \overline{\alpha}]$, a robust rescheduled timetable can still ensure that the actual operation of the train meets specific requirements.

Furthermore, we want to make the rescheduled timetable have varying degrees of robustness, thereby generating a set of high-quality robust alternatives from which dispatchers can select according to specific operational needs. To this end, $\underline{\alpha}$ and $\overline{\alpha}$ are also taken as decision variables. The following constraints are formulated.

Constraints of station dwell times:

For trains with dwell plans at stations, the dwell time of the train must exceed the specified operational time to accommodate passenger boarding and alighting. If trains can operate precisely as the rescheduled time, it can be easily formulated as $d_{k,i} - a_{k,i} \geq d^*_{k,i} - a^*_{k,i} (\forall i \in S;\ \forall k \in ST_i)$, like most existing research. Considering the deviation Δa between the actual arrival time of trains during operation and the rescheduled time $a_{k,i}$, the rescheduled timetable needs to ensure that, even under the impact of $\forall \Delta a \in [\underline{\alpha}, \overline{\alpha}]$, the actual operation of the train still meets the dwell time requirements. Hence, it is modelled as constraint (1).

In constraint (1), $\gamma^1_{\underline{\alpha}, \overline{\alpha}}$ is a binary (0-1) variable used to determine whether $\forall \Delta a \in [\underline{\alpha}, \overline{\alpha}]$ would cause the actual operation of the train to violate constraint: $d_{k,i} - a_{k,i} \geq d^*_{k,i} - a^*_{k,i} (\forall i \in S;\ \forall k \in ST_i)$. When $[\underline{\alpha}, \overline{\alpha}]$ includes trains arriving later than scheduled, $\gamma^1_{\underline{\alpha}, \overline{\alpha}}$ is set to 1. This is because a train arriving later than planned will shorten the train's dwell time, leading to the possibility that the train may fail to meet the minimum dwell time requirements if it departs as per the rescheduled timetable, as illustrated in Fig.2(b). Consequently, $\gamma^1_{\underline{\alpha}, \overline{\alpha}}$ is set to 1 to account for the negative impact that Δa might have on implementing a rescheduled timetable. If $[\underline{\alpha}, \overline{\alpha}]$ only involves scenarios where trains arrive earlier than scheduled, $\gamma^1_{\underline{\alpha}, \overline{\alpha}}$ is taken as 0, since early arrivals extend the dwell time, making it easier for the actual operation to meet the dwell time requirements, as shown in Fig.2(a). Thus, the impact of $\forall \Delta a \in [\underline{\alpha}, \overline{\alpha}]$ on the constraint (1) can be disregarded, and the constraint (1) can be simplified to their deterministic form: $d_{k,i} - a_{k,i} \geq d^*_{k,i} - a^*_{k,i} (\forall i \in S;\ \forall k \in ST_i)$.

For trains that directly pass through stations, the departure time is the same as the arrival time and is not affected by Δa, ensured by constraint (2).

$$d_{k,i} - \left(a_{k,i} + \gamma^1_{\underline{\alpha}, \overline{\alpha}} * \Delta a\right) \geq d^*_{k,i} - a^*_{k,i} \quad \forall \Delta a \in [\underline{\alpha}, \overline{\alpha}]; \forall i \in S; \forall k \in ST_i \tag{1}$$

$$d_{k,i} = a_{k,i} \quad \forall i \in S; \forall k \in PT_i \tag{2}$$

Constraints of minimum section running times:

In favor of safety reasons, the train running time on section i-1 should not fall below the minimum time r_{i-1}. To ensure that the actual operation of the train still exceeds the minimum section running time, even under the impact of $\forall \Delta a \in [\underline{\alpha}, \overline{\alpha}]$, the constraint is modeled as constraint (3).

In constraint (3), besides $\gamma^1_{\underline{\alpha}, \overline{\alpha}}$, $\gamma^2_{\underline{\alpha}, \overline{\alpha}}$ is also used to determine whether $[\underline{\alpha}, \overline{\alpha}]$ includes cases that cause the train's actual operation to violate the deterministic equivalent form of constraint (3): $d_{k,i} - a_{k,i} \geq r_{i-1} (\forall i \in S \setminus \{1\};\ \forall k \in T)$. The value of $\gamma^2_{\underline{\alpha}, \overline{\alpha}}$ is exactly the opposite of that of $\gamma^1_{\underline{\alpha}, \overline{\alpha}}$. When $[\underline{\alpha}, \overline{\alpha}]$ encompass scenarios where trains arrive earlier than scheduled, as illustrated by the green solid line in Fig.2(a),

$\gamma^2_{\underline{\alpha},\overline{\alpha}}$ should be set to 1. Because early arrivals often result in trains operating at speeds higher than planned, reducing the section running time, this may consequently fail to meet the minimum section running time constraints. Therefore, the impact of $\forall \Delta a \in [\underline{\alpha},\overline{\alpha}]$, including early arrivals, must be considered in the constraint (3) to ensure the robustness of the timetable. If $[\underline{\alpha},\overline{\alpha}]$ merely involves trains arriving later than scheduled, $\gamma^2_{\underline{\alpha},\overline{\alpha}}$ is set to 0. A train arriving later than planned extends the section running time of the train, making the actual operation to meet the minimum section running time constraints easier, as Fig.2(b) shows. There is no need to consider the impact of $\forall \Delta a \in [\underline{\alpha},\overline{\alpha}]$ on the constraint (3).

$$\begin{aligned}\left(a_{k,i}+\gamma^2_{\underline{\alpha},\overline{\alpha}}*\Delta a\right)-\left[d_{k,i-1}+\left(1-x^*_{k,i-1}\right)*\gamma^1_{\underline{\alpha},\overline{\alpha}}\cdot\Delta a\right]\\ \geq r_{i-1}\quad \forall \Delta a\in[\underline{\alpha},\overline{\alpha}];\forall i\in S\backslash\{1\};\forall k\in T\end{aligned} \tag{3}$$

Constraints of velocity limitation:

Perturbations impose extra limitations on train velocity, increasing the minimum running times for the impacted sections, represented as r^{ex}_{i-1}. In a manner akin to the constraints of minimum section running times, this extra speed restriction is formulated as constraint (4).

$$\begin{aligned}\left(a_{k,i}+\gamma^2_{\underline{\alpha},\overline{\alpha}}*\Delta a\right)-\left[d_{k,i-1}+\left(1-x^*_{k,i-1}\right)*\gamma^1_{\underline{\alpha},\overline{\alpha}}\cdot\Delta a\right]\\ \geq r^{ex}_{i-1}\quad \forall \Delta a\in[\underline{\alpha},\overline{\alpha}];\forall i\in S\backslash\{1\};\forall k\in T\end{aligned} \tag{4}$$

Constraints of arrival headway:

To prevent train collisions, there should be a safe headway H_{min} between two consecutive trains arriving at the same station in the same direction. If the preceding train arrives later than scheduled, or the following train arrives earlier than planned, it could result in the actual headway between the arrivals of two consecutive trains being less than the minimum headway H_{min}. Therefore, when constructing the model, it is necessary to consider the impact of deviations in executing the rescheduled timetable, leading to constraint (5).

$$\begin{aligned}\left(a_{k+1,i}+\gamma^2_{\underline{\alpha},\overline{\alpha}}*\Delta a\right)-\left(a_{k,i}+\gamma^1_{\underline{\alpha},\overline{\alpha}}*\Delta a\right)\geq H_{min}\\ \forall \Delta a\in[\underline{\alpha},\overline{\alpha}];\forall i\in S;\forall k\in T\backslash\{m\}\end{aligned} \tag{5}$$

Two consecutive trains departing from the same station also need to keep a safe headway H_{min}, just like arrival. Given the negative impact of Δa on the departure times of trains without dwell plans, constraint (6) guarantee that trains can still maintain a safe headway H_{min} despite the influence of $\forall \Delta a \in [\underline{\alpha},\overline{\alpha}]$.

$$\begin{aligned}\left[d_{k+1,i}+\left(1-x^*_{k+1,i}\right)*\gamma^2_{\underline{\alpha},\overline{\alpha}}*\Delta a\right]-\\ \left[d_{k,i}+\left(1-x^*_{k,i}\right)*\gamma^1_{\underline{\alpha},\overline{\alpha}}*\Delta a\right]\geq H_{min}\\ \forall \Delta a\in[\underline{\alpha},\overline{\alpha}];\forall i\in S;\forall k\in T\backslash\{m\}\end{aligned} \tag{6}$$

Constraints of track capacity:

With a limited number of tracks available at stations, it is unavoidable for different trains to occupy the same track. To prevent trains from colliding, when train g runs before train k, and both need to occupy the same track p at station i, the headway between the arrival time $a_{k,i}$ of train k and the departure time $d_{g,i}$ of train g must be at least h_{min}. However, if the preceding train departs later than its scheduled time or the following train arrives earlier than its scheduled time, the actual headway between trains will be shorter than the rescheduled headway. In order to make the rescheduled headway can resist $\forall \Delta a \in [\underline{\alpha},\overline{\alpha}]$, constraint (7) are constructed.

$$\begin{aligned}\left(a_{k,i}+\gamma^2_{\underline{\alpha},\overline{\alpha}}*\Delta a\right)-\left[d_{g,\,i}+\left(1-x^*_{g,\,i}\right)\gamma^1_{\underline{\alpha},\overline{\alpha}}*\Delta a\right]\geq h_{min}\\ -\left(3-y^*_{g,k,i}-Z^*_{k,p,i}-Z^*_{g,p,i}\right)*M\\ \forall \Delta a\in[\underline{\alpha},\overline{\alpha}];\forall i\in S;\forall k,g\in T:k>g;\forall p\in C_i\end{aligned} \tag{7}$$

Constraints of interval boundary:

The values for the interval's upper and lower bounds should lie between the maximum and minimum values of Δa as found in historical statistical data.

$$LB\leq\underline{\alpha}\leq\overline{\alpha}\leq UB \tag{8}$$

where *LB* is the minimum value of Δa as found in historical statistical data, and *UB* is the maximum value of Δa as found in historical statistical data.

Adding the following constraint eliminates redundant combinations that yield identical results and facilitates the computational process, ultimately improving solution efficiency. For example, when both $\underline{\alpha}$ and $\overline{\alpha}$ are greater than 0, the solution also has the capability to resist $\forall \Delta a \in [0,\overline{\alpha}]$.

$$\underline{\alpha}\leq 0\leq\overline{\alpha} \tag{9}$$

Constraints for interval-determining variables of random variable:

In the explanations of constraints (1) and (3), we detail how the binary variables $\gamma^1_{\underline{\alpha},\overline{\alpha}}$ and $\gamma^2_{\underline{\alpha},\overline{\alpha}}$ are determined based on $[\underline{\alpha},\overline{\alpha}]$. Given that the impact of trains arriving earlier or later than the rescheduled timetable on each constraint is different, it is essential to determine whether to incorporate Δa into the constraints according to the specific situation of the selected interval $[\underline{\alpha},\overline{\alpha}]$. Their relationship can be expressed by the following formulations.

$$\overline{\alpha}>-M*\left(1-\gamma^1_{\underline{\alpha},\overline{\alpha}}\right) \tag{10}$$

$$\overline{\alpha}\leq M*\gamma^1_{\underline{\alpha},\overline{\alpha}} \tag{11}$$

$$\underline{\alpha}<\left(1-\gamma^2_{\underline{\alpha},\overline{\alpha}}\right)*M \tag{12}$$

$$\underline{\alpha}\geq -M*\gamma^2_{\underline{\alpha},\overline{\alpha}} \tag{13}$$

C. Objective functions

Objective 1 - Minimizing the delay propagation risk (shown in Fig.4): The number of adjustments required for a rescheduled timetable is inherently difficult to formulate within a mathematical model, as it emerges from the dynamic process of timetable failure and iterative updating while applying in the stochastic environment. Nevertheless, during the course of our study, we observed a strong correlation between the

delay propagation risk and the number of adjustments, which is illustrated in Fig.5. Specifically, rescheduled timetables with lower delay propagation risk tend to be more robust, with a lower likelihood of failure and fewer subsequent adjustments required. Therefore, we adopt delay propagation risk as a surrogate metric to quantify differences in the number of adjustments among candidate rescheduled timetables. This objective function can be formulated as follows:

$$\min_{\{a_{k,i}d_{k,i},\underline{\alpha},\overline{\alpha}\}} Z_1 = \int_{\overline{\alpha}}^{UB} f(\Delta a)\, d\Delta a \tag{14}$$

where $f(\ )$ stands for the probability distribution function of Δa and UB is the maximum value of Δa in historical statistical data.

Objective 2 - Minimizing the total delay in the rescheduled departure and arrival times: Rescheduling typically aims to minimize the total delay time of trains in order to mitigate the adverse effects of unexpected perturbations on railway operations. The formulation of this objective function is shown in equation (15).

$$\min_{\{a_{k,i}d_{k,i},\underline{\alpha},\overline{\alpha}\}} Z_2 = \sum_{i\in S}\sum_{k\in T}\left[\left(a_{k,i}-a^*_{k,i}\right)+\left(d_{k,i}-d^*_{k,i}\right)\right] \tag{15}$$

V. Solution Algorithm

To efficiently approximate the Pareto front of the multi-objective optimization model, an MP-NSGA-II is proposed. Section V-A introduces the core techniques and concepts of MP-NSGA-II. Section V-B details the algorithm's implementation steps.

NSGA-II is one of the most widely used algorithms for solving the Pareto-optimal set (POS) of multi-objective optimization problems, demonstrating strong performance and broad applicability across various fields [38]. Therefore, we design our algorithm based on NSGA-II. The high performance of NSGA-II relies on its evolution mechanisms, mainly including the fast nondominated sorting, crowding distance sorting, and nondominated ranking-based elite preservation strategy. The detailed procedure can be found in the literature [30].

TTR typically requires decision-making to be completed within a short time frame. This also requires the algorithm to generate higher-quality Pareto front as quickly as possible, while maintaining stable and reliable performance. Although the traditional NSGA-II algorithm demonstrates strong global exploration capabilities, it exhibits relatively weak local search ability. This restricts the ability of NSGA-II to improve the Pareto front during the evolutionary process, making it difficult to approximate the true Pareto front within a very limited amount of time. To enhance the applicability of the NSGA-II algorithm to TTR problem, we propose an MP-NSGA-II.

A. Key Concepts and Techniques for MP-NSGA-II

To facilitate a better understanding of the proposed algorithm, the key concepts and techniques are introduced and explained in this section.

Chromosome structures

To simplify the model-solving process, a hierarchical optimization strategy for decision variables is adopted. A distinctive feature of the proposed model is that once the values of $\underline{\alpha}$ and $\overline{\alpha}$ are specified, the value of $\gamma^j_{\underline{\alpha},\overline{\alpha}}$ can be determined accordingly. The sub-model composed of objective function Z_2 and constraints (1)–(7) becomes a relatively simple robust optimization problem, which can be solved directly. Thus, a robust rescheduled timetable with the minimum total delay that can withstand the arrival deviation in the specific interval $[\underline{\alpha},\overline{\alpha}]$ is obtained, and the objective Z_1 can also be calculated by equation (14). Based on the above ideas, it is essential to design an algorithm to find appropriate combinations of $\underline{\alpha}$ and $\overline{\alpha}$. To this end, the chromosomes are encoded as vectors composed of two-dimensional arrays X = [L, U], where each array element represents the values of $\underline{\alpha}$ and $\overline{\alpha}$.

Dividing the search space

The complete search space is divided into several search regions based on the value ranges of $\underline{\alpha}$ and $\overline{\alpha}$. Within each search region, multiple individuals are distributed, allowing the algorithm to efficiently explore the entire search space. This strategy also could ensure the stability of the optimization results to a certain extent.

Crossover and mutation procedure

Crossover is performed by randomly selecting two parent individuals $X1$ = [$L1$, $U1$] and $X2$ = [$L2$, $U2$] generating two child individuals $cX1$ = [$L1$, $U2$] and $cX2$ = [$L2$, $U1$] by exchanging elements within their chromosome arrays.

Mutation is a mechanism for maintaining the diversity of offspring populations. We randomly select an element from a parent's chromosome and replace it, thereby generating a new child individual. For example, the child individual cX = [NL, U] is created by replacing $\underline{\alpha}$ value in the chromosome of parent individual X = [L, U] with a new value.

Multiple cooperative populations

To achieve a more effective balance between global exploration and local exploitation in the search space, we distribute two distinct yet complementary populations: a local search population and an auxiliary local search population. By strategically distributing these populations and leveraging their collaborative interaction, the algorithm is able to maintain exploration diversity while intensifying search efforts in promising regions, thereby enhancing overall optimization performance.

1) Local Search Population: The local search population is confined to perform exploration within a fixed subregion of the overall search space. It is further partitioned into multiple subpopulations, and each search region is assigned a subpopulation, enabling continuous local search within that area. The mutation operations for individuals in this population are restricted to the decision variable bounds specific to their respective region. Regions where consistently well-performing subpopulations are located are more likely to contain potential Pareto-optimal solutions and therefore merit intensified local exploitation. To this end, we introduce an auxiliary local search popu-

lation to enhance the algorithm's exploitation capability in such promising regions.

2) Auxiliary Local Search Population: This is a population that can perform stochastic movement across the search space, which is designed to enhance the local exploitation ability of the algorithm. It is divided into several subpopulations. Initially, we distributed an auxiliary local search subpopulation in each search region. Inspired by the pheromone mechanism in Ant Colony Optimization (ACO), the performance of local search subpopulations is transformed into pheromone to guide the auxiliary local search population. As pheromones evolve during the iteration process, the auxiliary local search subpopulation is more likely to emerge in the region where the search results continue to perform well, assisting the local search subpopulation in that region to exploit. The mutation operations for individuals in this population are restricted to the decision variable bounds specified by their assigned region.

External Archive Technique combined with Neighborhood Search

External archive is a common technique used in many multi-objective algorithms to store the elite solutions found in each generation [39]. Since these archived elite solutions often exhibit close proximity to the true POS, we incorporate a neighborhood search technique to exploit their local surroundings and strengthen the algorithm's local exploitation ability. The Algorithm 1 shows the procedure of the neighborhood search technique.

Algorithm 1 Neighborhood Search

Input: archived elite individuals X = [L, U]
Output: child individuals NX

Let Qx be the number of archived elite individuals
for $i = 1 \rightarrow Qx$ **do**
  generate two random number R(i) and B(i), $0 \leq R(i) \leq 1$, B(i) =1 or -1.
  if $R(i) \geq 0.5$ **then**
    $NU1$ = U+ $B(i)$; $NU2$ = min{$NU1$, UB};
    $NU3$ = max{$NU2$, 0};NX = [L, $NU3$]
  else
    $NL1$ = L+ B(i); $NL2$ = max{$NL1$, LB};
    $NU3$ = min{$NL2$, 0};NX = [$NL3$, U]
  end if
end for

Information Feedback Mechanism

To enhance the search efficiency, high-performing individuals generated in each iteration are systematically fed back into the local search subpopulation in each region. By incorporating these elite solutions into the next iteration, the algorithm is adaptively steered toward promising areas of the search space, increasing the probability of discovering similar or improved solutions in their neighborhood.

B. Implementation steps of MP-NSGA-II

The detailed implementation steps of the MP-NSGA-II are as follows.

***Step 1*:** Initialization. $EA = \emptyset$. Divide the entire search space into u regions $PS_v, v = 1, 2, .., u$.

***Step 2*:** Initialize the local search population (LS) and the auxiliary local search population (AS). Both LS and AS are subdivided into u subpopulations, each assigned to a specific search region. $LS = \{LS_v \mid v = 1, 2, \ldots, u\}, AS = \{AS_v \mid v = 1, 2, \ldots, u\}$.

***Step 3*:** Initialize pheromones. $A = LS \cup AS$. Rank all individuals in the set A by non-dominated sorting and crowding distance sorting. Let R_q represent the rank of the q-th individual in the set A. For each search region PS_v, the potential POS score SC_v is computed according to the following equation:

$$SC_v = \frac{1}{\sum_{q \in PS_v} R_q} \quad v = 1, 2, \ldots, u \tag{16}$$

The pheromone of each search region PS_v, denoted as IN_v, is initialized to SC_v.

***Step 4*:** Main loop:

while *Termination condition is not met* **do**

***Step 4.1*:** Update local search subpopulations: Generate the child subpopulation by applying crossover and mutation operators to LS_v. After merging the parent and child subpopulations, apply non-dominated sorting to rank the combined set. Replace all original members of LS_v by the top Q_v individuals in the combined set.

***Step 4.2*:** Update pheromones: Rank all individuals in LS by non-dominated sorting and crowding distance sorting. Calculate SC_v by equation (16). $SC = \{SC_v \mid v = 1, 2, \ldots, u\}$. $IN_v = (1-\rho) * IN_v + \Delta IN_v$, where ρ is the evaporation rate and ΔIN_v is the quantity of pheromone laid on search region PS_v by LS_v. ΔIN_v is given by:

$$\Delta IN_v = \begin{cases} SC_v & \text{if } SC_v \text{ is the largest element in } SC \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

***Step 4.3*:** Update auxiliary local search subpopulations: calculate the selection probability P_v of LS_v using the following formula:

$$P_v = \frac{IN_v}{\sum_{v=1}^{u} IN_v} \quad v = 1, 2, \ldots, u \tag{18}$$

Choose a local search subpopulation LS_w using the pseudo-random proportional rule [40]. $AN_v = AS_v \cup LS_w$. Generate the child subpopulation by applying crossover and mutation operators to AN_v. After merging the parent and child subpopulations, apply non-dominated sorting to rank the combined set. Replace all original members of AN_v with the top Q_v individuals in the combined set.

***Step 4.4*:** Apply non-dominated sorting to ALL to extract the subset F containing all non-dominated individuals. $NewF = EA \cup F$. Apply non-dominated sorting to $NewF$ to extract the subset $F1$ containing all non-dominated individuals. EA = $F1$

***Step 4.5*:** Perform a neighborhood on EA using Algorithm 1 to obtain NEA.

***Step 4.6*: $F2 = EA \cup NEA$**. Apply non-dominated sorting to *F2* to extract the subset *F3* containing all non-dominated individuals. *EA = F3*.

***Step 4.7*:**Information feedback: feedback the subset F_l of individuals from *F3* to each local search subpopulation. $RT_v = \{F_l \mid (F_l \in PS_v) \wedge (F_l \in F3)\}$, $LS_v = LS_v \cup RT_v, v = 1, 2, .., u$.

end while

C. Complexity analysis of MP-NSGA-II

We analyze the worst-case time complexity using the following notations, and the results of the algorithm complexity analysis are summarized in Table III.

- u: number of partitioned search regions.
- Q: fixed size of each subpopulation (LS_v and AS_v).
- $N = u \cdot Q$: total number of individuals in the local search population.
- MS: number of objectives.
- L: length of the decision vector (individual encoding).
- G: maximum number of generations.
- FC: computational cost of a single fitness evaluation.
- ES: size of the external archive EA.

TABLE III
RESULTS OF COMPLEXITY ANALYSIS OF MP-NSGA-II.

Steps	Complexity
Steps 1–3 (Initialization)	$O(N \cdot FC + MS \cdot N^2)$
Step 4.1	$O(MS \cdot N \cdot Q + N \cdot L + N \cdot FC)$
Step 4.2	$O(MS \cdot N^2)$
Step 4.3	$O(MS \cdot N \cdot Q + N \cdot L + N \cdot FC)$
Step 4.4	$O(MS \cdot N^2)$
Step 4.5	$O(ES \cdot FC)$
Step 4.6	$O(MS \cdot ES^2)$
Step 4.7	$O(N)$
Per-generation	$O(MS \cdot N^2 + N \cdot L + N \cdot FC)$
Overall complexity	$O\big(G \cdot (MS \cdot N^2 + N \cdot L + N \cdot FC)\big)$

VI. NUMERICAL EXPERIMENTS

The proposed method is validated using data from the Elizabeth Line in London. Section VI-A introduces some basic experimental settings. We display and analyze the experimental results from section VI-B to section VI-D.

A. Experiments basic setup information

The experimental validation is conducted from two perspectives: the proposed multi-objective optimization model and the effectiveness of the MP-NSGA-II. The experiments are based on the realistic dataset from the Elizabeth Line in London. We use the westbound timetable of the Elizabeth Line to test our method.

To simulate the impact of varying perturbation intensities on different line types, two levels of perturbation (strong and weak) are separately set to affect the main and branch lines of the Elizabeth Line, thereby generating four distinct experimental scenarios. Assuming that the arrival deviations follow a log-normal distribution. This assumption is motivated by the empirical observation that small deviations occur frequently while large deviations occur with lower probability, exhibiting right-skewed characteristics. As documented in the literature, the log-normal distribution is better able to capture the distributional characteristics of arrival deviations [41]. Nevertheless, our method is not restricted to the log-normal distribution. Substituting it with an alternative distribution function only affects the calculation of Z_1 using different functions; the proposed model and algorithm remain applicable. The distribution function varies slightly under different levels of perturbation intensity : (1) under weak perturbation, the expected deviation $\mu = 1.8$s, the standard deviation $\sigma = 0.3$s and the location parameter c = -5.5s; (2) under strong perturbation, the expected deviation $\mu = 2.5$s, the standard deviation $\sigma = 0.5$s and the location parameter c = -9s. The detailed information of the four test scenarios is organized in TABLE IV.

All codes are implemented in Matlab R2023a on Windows 10 and run on the computer with processor 13th Gen Intel(R) Core(TM) i9-13900K CPU 3.00 GHz with 128 GB RAM memory. The commercial solver GUROBI 11.0.2 is employed to assist in solving the multi-objective optimization model, with all solver parameters set to their default values.

B. Robustness of rescheduled timetables against arrival deviations within a specific interval

In this section, we want to examine whether the constructed constraints can endow the rescheduled timetable with robustness against arrival deviations within a specific interval. Robustness is reflected by Constraint Violation Percentage (CVP). A lower CVP value indicates better robustness. Each indicator is calculated based on 50 repeated independent experiments. The detailed explanation of CVP indicators is as follows.

CVP: an Added Deviation Rescheduled Timetable (ADRT) is obtained by directly adding arrival deviation to the rescheduled timetable. Check whether the ADRT meets the specified requirements, specifically the constraints (19)-(24) in the APPENDIX. Then, CVP can be calculated as follows.

$$\mathrm{C\ V\ P} = \frac{\text{the number of constraint which ADRT does not satisfy}}{\text{the number of all constraint}} \times 100\%$$

In each experimental scenario, the arrival deviations interval $[\underline{\alpha}, \overline{\alpha}]$ is set as TABLE V. The robust timetables is obtained by solving the model formulated with objective function Z_2 and constraints (1)-(7) and (10)-(13). Three different robust timetables are obtained respectively by solving the model. Test the timetable's robustness against arrival deviations within its predefined interval. The arrival deviations within the timetable setting interval are generated in four test scenarios. Calculate CVP of robust rescheduled timetables under the influence of the arrival deviations within the setting interval, and the results are presented in TABLE VI.

TABLE VI shows that, across all testing scenarios, the maximum CVP of 50 timetables' robustness (against arrival deviations within timetable predefined intervals) test experiments is 0. Since the robust timetable generated by our method can ensure that the actual operation of the train still meets the constraints even if it is influenced by arrival

TABLE IV
BASIC CONFIGURATION INFORMATION FOR THE FOUR EXPERIMENTAL TEST SCENARIOS

Scenario number	S1	S2	S3	S4
Speed limit areas	WOLWXR CUSTEBT CUSTMHS	WOLWXR CUSTEBT CUSTMHS CUSTWBT CANWHRF	WCHAPXR WCHAVRD LIVSTLL	WCHAPXR WCHAVRD LIVSTLL FRNDXR FRNDFST
Number of affected trains	6	11	12	23
Speed limit time	18:00-18:30	18:00-19:00	18:00-18:30	18:00-19:00
Extended running time(s)	10	30	10	30
Perturbation level	Weak	Strong	Weak	Strong

TABLE V
THE INTERVAL BOUNDARY OF ARRIVAL DEVIATION SET IN THE MODEL TO SOLVE ROBUST RESCHEDULED TIMETABLES

Interval Index	**Lower bounds of deviations interval**	**Upper bounds of deviations interval**
1	$\phi^{-1}(40\%)$	$\phi^{-1}(60\%)$
2	$\phi^{-1}(20\%)$	$\phi^{-1}(80\%)$
3	$\phi^{-1}(0)$	$\phi^{-1}(99.9\%)$

Symbol descriptions: $\phi^{-1}(\cdot)$ represents the inverse function of the cumulative distribution function (CDF) of arrival deviation.

deviation with specific intervals, the timetable remains valid without needing adjustments. As a result, in the robustness experiments conducted under arrival deviations within the timetable predefined interval, the maximum values of CVP are 0.

The experiment's findings confirm that the robust timetables generated using our proposed model can effectively resist arrival deviations within a specific interval. It also means that when trains operate according to our robust timetable, even if there is a certain range of deviation between the actual arrival time and the rescheduled plan, our timetable can ensure that the actual operation of the trains still meets specific operational requirements. Thus, it can continue to be used without any adjustments.

C. Verification of the effectiveness of the multi-objective optimization model

The verification experiments for the model's effectiveness are divided into two aspects: 1) whether the constructed multi-objective model can yield robust rescheduled timetables with fewer adjustment numbers; and 2) whether the Pareto front of the model accurately reflects the trend of adjustment numbers. Specifically, the second aspect is a statistical verification of the correlation between the delay propagation risk (Z_1) and adjustment numbers. The more detailed explanation about adjustment numbers is as follows.

Adjustment numbers: The number of adjustments towards the failure rescheduled timetable during its execution phase. When applied in practice, rescheduled timetables may become unworkable under the impact of arrival delays if they fail to guarantee that subsequent train operations meet specific requirements, as shown in Fig.2(b). When faced with such circumstances, it is necessary to shift the subsequent train timetable to ensure it remains functional. In uncertain environments, such timetable-shifting adjustments often need to be performed repeatedly. The more robust the timetable, the less frequent the adjustments will be.

For each test scenario, we enumerate all possible combinations of values for $\underline{\alpha}$ and $\overline{\alpha}$. For each combination, a robust rescheduled timetable with the minimum total delay that can withstand the arrival deviation in the specific interval $[\underline{\alpha}, \overline{\alpha}]$ is obtained by solving the sub-model composed of constraints (1)–(13) and the objective function Z_2. Then, the objective function Z_1 is calculated for each combination based on the corresponding formula (14). Then, based on the probability distribution function of the uncertain arrival deviations in each scenario, we randomly generated 50 test cases to measure the adjustment numbers required for these robust timetables. In each scenario, we select two POSs. The first POS is selected using the median of adjustment numbers in the 50 experiments and total delays of rescheduled timetables. The second POS is selected using the objective function Z_1 and total delays. TABLE VII presents the number of elements in each of the two POSs, as well as the number of common elements between them. For all timetables in the second type of POS, line charts are constructed using the total delay as the X-axis, with the standardized objective function Z_1 and the standardized median number of adjustment numbers across 50 experiments along the Y-axis, respectively. These charts are used to analyze the similarity in the trend patterns of the two indicators, which are presented in Fig.6.

According to TABLE VII, the number of elements in POS selected by Z_1 is slightly higher than that selected by adjustment numbers across four scenarios. The POS selected by adjustment numbers is fully contained within the POS selected by Z_1. Proportion of common elements in POS selected by Z_1 in scenarios 1–4 is 75%, 71%, 75%, and 84%, respectively, all exceeding 70%. Among them, scenario 4 exhibits the highest overlap, while the proportions in scenarios 1, 2, and 3 are relatively close, with scenario 2 being the lowest.

As shown in Fig.6, the trends of the objective function Z_1 and the adjustment numbers across the four scenarios are quite similar, though not entirely consistent, with some minor differences. Both indicators show a downward trend as total delays increase. The decline speed is faster in the early stage and gradually slows down in the later stage. This difference in change speed is most obvious in scenario 4. As total delays increase, adjustment numbers in the four scenarios decrease to 0 earlier than Z_1.

TABLE VI
CVP OF TIMETABLE ROBUSTNESS EXPERIMENTS AGAINST ARRIVAL DEVIATION WITHIN ITS PREDEFINED INTERVAL

Interval boundary set in solving robust rescheduled timetables		**Interval boundary of arrival deviations generated in robustness experiments**		**The max CVP of 50 independent repeated experiments**			
Low	**Up**	**Low**	**Up**	**S1**	**S2**	**S3**	**S4**
$\phi^{-1}(40\%)$	$\phi^{-1}(60\%)$	$\phi^{-1}(40\%)$	$\phi^{-1}(60\%)$	0	0	0	0
$\phi^{-1}(20\%)$	$\phi^{-1}(80\%)$	$\phi^{-1}(20\%)$	$\phi^{-1}(80\%)$	0	0	0	0
$\phi^{-1}(0)$	$\phi^{-1}(99.9\%)$	$\phi^{-1}(0)$	$\phi^{-1}(99.9\%)$	0	0	0	0

TABLE VII
NUMBER OF SOLUTIONS IN TWO PARETO SETS AND THEIR OVERLAP

Scenario Index	**S1**	**S2**	**S3**	**S4**
The number of elements in POS selected by adjustment numbers	6	22	6	26
The number of elements in POS selected by Z_1	8	31	8	31
The number of common elements between two POSs	6	22	6	26
Proportion of common elements in POS selected by Z_1	75%	71%	75%	84%

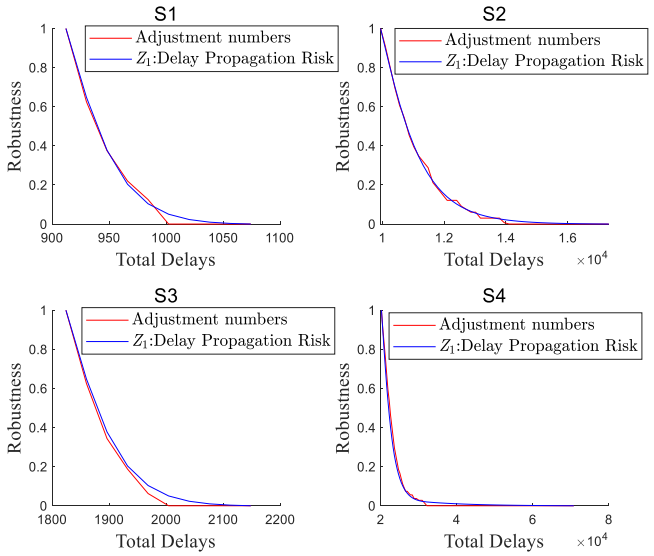


Fig. 6. Trends of adjustment numbers and Z_1 with respect to total delays.

There is a strong correlation between adjustment numbers and the objective function Z_1: the delay propagation risk. The greater the delay propagation risk, the higher the probability that the rescheduled timetable will fail under the influence of uncertain arrival deviations. Correspondingly, the rescheduled timetable needs more frequent adjustments, so adjustment numbers are also greater. Therefore, the delay propagation risk can effectively reflect adjustment numbers. However, the two indicators are not entirely equivalent, and the method for calculating adjustment numbers involves a degree of randomness. As a result, it is inevitable that there will be differences in the trends of these two indicators and the corresponding Pareto-optimal solutions. In each scenario, test cases for adjustment numbers are randomly generated based on the probability distribution function of arrival deviations. This leads to a relatively lower proportion of cases with longer delays, so the difference in adjustment numbers test results for some more robust rescheduled timetables may not be obvious. Under the assumption that arrival deviations follow a log-normal distribution, the objective function Z_1 quantifies the risk that a delay may propagate at a given station under the influence of various potential future arrival deviations. However, once an actually occurring deviation exceeds the disturbance-absorption capability of the robust rescheduled timetable, determining whether a delay will propagate to the next station and to subsequent trains requires the consideration of more complex factors, such as the slack times added to the minimum running times in the subsequent section and the buffer times added to the minimum headways between consecutive trains. Such detailed cascading and multi-train delay propagation mechanisms are not explicitly represented in the current model, which may account for the discrepancies observed between the two indicators. To enhance the robustness of the rescheduled timetable against uncertain arrival deviations, buffer time needs to be added to the rescheduled timetable, but this will also cause more delays. As a result, Z_1 and adjustment numbers both exhibit a decreasing trend as total delays increase. According to empirical statistical distribution, small deviations occur with higher probability. Therefore, introducing a small amount of buffer time can significantly reduce Z_1 and adjustment numbers. However, the longer the delay, the lower the probability of occurrence. Excessive buffer time does not significantly further reduce Z_1 and adjustment numbers.

The experimental results demonstrate the effectiveness of the proposed multi-objective optimization model. The proposed objective function Z_1: delay propagation risk can indeed serve as a good indirect reflection of adjustment numbers. It assists dispatchers in generating rescheduled timetables that require fewer adjustments and reflects the trend of adjustment numbers changing with the total delays. The Pareto fronts represented by the blue solid lines in Fig.6 not only provide dispatchers with a set of high-quality alternative robust rescheduled timetables, but also visually illustrate the trade-off relationship between the two objective functions across different robust rescheduled timetables. This approach aids dispatchers in balancing the conflict between the two objective functions of robust rescheduled timetables. Further, this helps dispatchers better weigh and compare various robust rescheduled timetables, enabling them to select or formulate a robust timetable more aligned with real-world rescheduling scenarios based on actual rescheduling requirements and their own experience. This is precisely the decision support tool we

need in this study.

D. Comparative Analysis Between the MP-NSGA-II and Other Algorithms

In this section, the proposed MP-NSGA-II is compared with NSGA-II [30], INSGA-II [36], and MOGWO [42]. The maximum iteration for each algorithm is set to 35 to observe their performance in approximating the true Pareto front. Two widely adopted performance indicators, Hypervolume (HV) and Inverted Generational Distance (IGD), are used to evaluate and measure the quality of the Pareto front obtained by each algorithm during the optimization process [39]. The two indicators measure the convergence and diversity of the solution sets. A larger HV indicates better convergence and diversity, while a smaller IGD, which is based on the distance between the obtained solution set and the true POS, is preferred. When the algorithm finds the complete POS, the algorithm is terminated early. All the algorithms are run 10 times on the four test scenarios. The statistical results of these 10 runs are provided in Fig.7-Fig.10.

The crossover and mutation probabilities of NSGA-II, INSGA-II and MP-NSGA-II are set to 0.45. For MP-NSGA-II, the evaporation rate ρ is set to 0.3. For MOGWO, grid inflation parameter, alpha = 0.1; number of grids per each dimension, nGrid = 10; leader selection pressure parameter, beta = 4; extra (to be deleted) repository member selection pressure, gamma = 2. For all four algorithms, the population size is set to 16 under weak perturbation scenarios and 36 under strong perturbation scenarios. For MP-NSGA-II, the complete search space is divided into several search regions based on the value ranges of $\underline{\alpha}$. In weak perturbation scenarios, the search space is divided into two regions: [-5.5, -3] and (-3, 0]. In strong perturbation scenarios, the search space is divided into three regions: [-9, -6], (-6, -2], and (-2, 0]. All individuals in the MP-NSGA-II population are equally divided into the local search population and the auxiliary local search population. Both the local search population and the auxiliary local search population are divided into several subpopulations, with the number of subpopulations equal to the number of regions. The number of individuals in each sub-population is equal.

To enable a fairer comparison among the four algorithms, both the average quality of the Pareto front obtained during the algorithm iteration process and the corresponding computation time are jointly analyzed and presented in Fig.7 and Fig.8. MOGWO can achieve a higher-quality Pareto front with less computation time. Our algorithm typically surpasses MOGWO after several iterations, demonstrating a certain advantage, with this phenomenon being most evident in scenario 4. While the performance of HV and IGD in each iteration of MP-NSGA-II initially surpasses that of MOGWO, MOGWO's faster iteration time allows it to enhance these indices in a short amount of time by performing multiple iterations. However, once the quality of the Pareto front reaches a certain level, it is essential for the algorithm to possess strong local search capabilities to accurately locate solutions within the true POS, in order to further improve the quality of the front. The local search ability of the MOGWO algorithm gradually

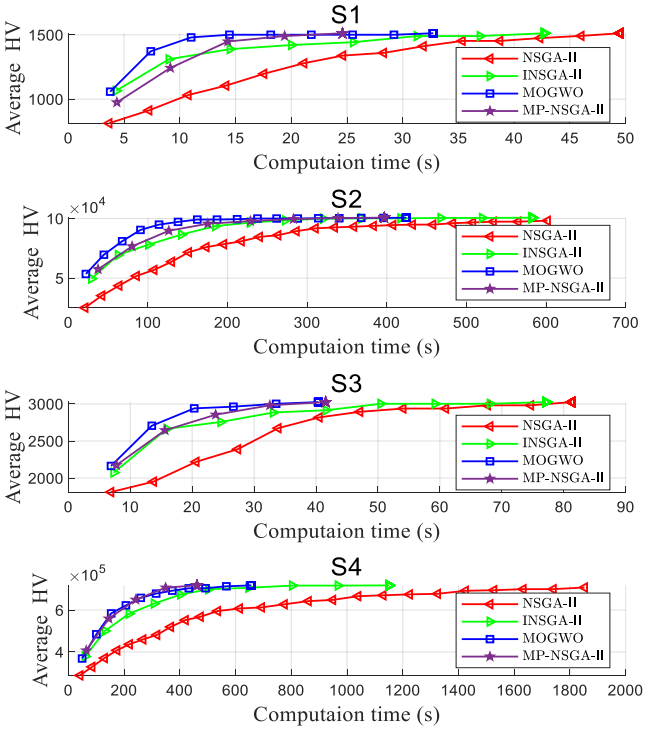


Fig. 7. Average HV and computation time during algorithm Iterations.

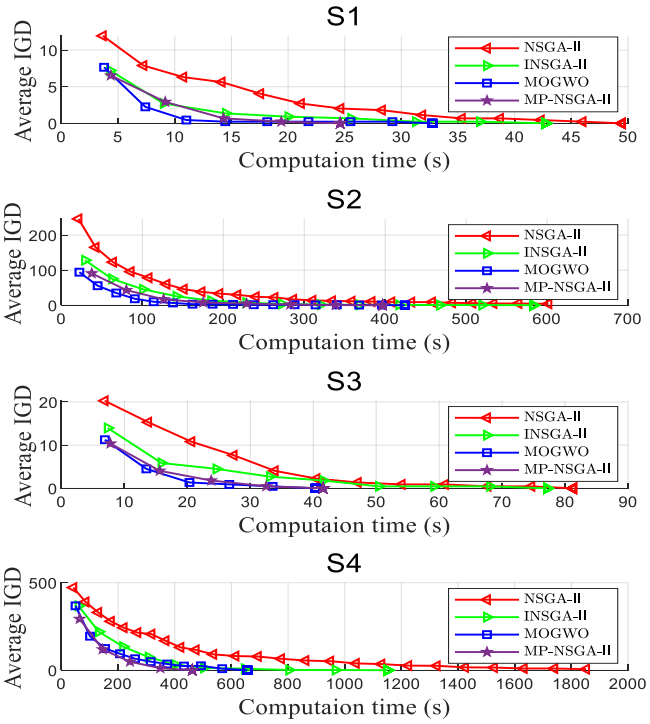


Fig. 8. Average IGD and computation time during algorithm Iterations.

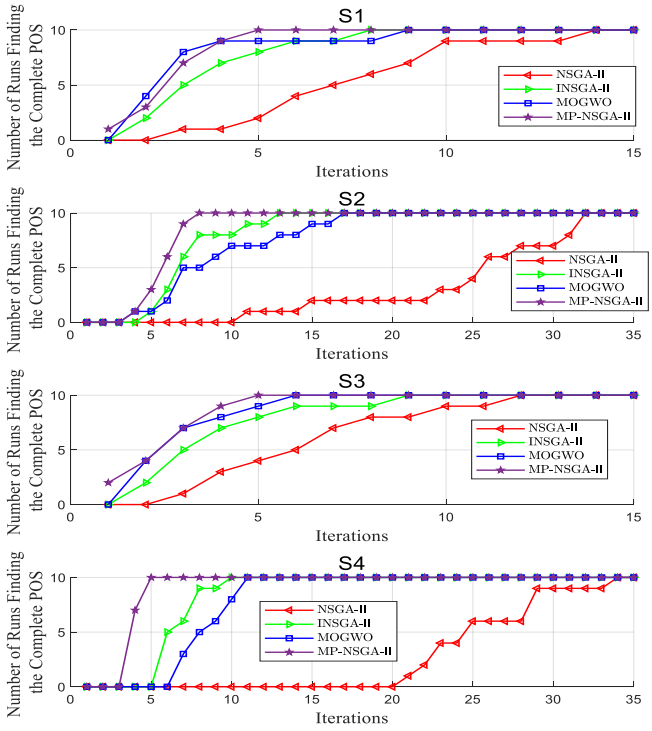


Fig. 9. Iterations to obtain the complete POS for four algorithms.

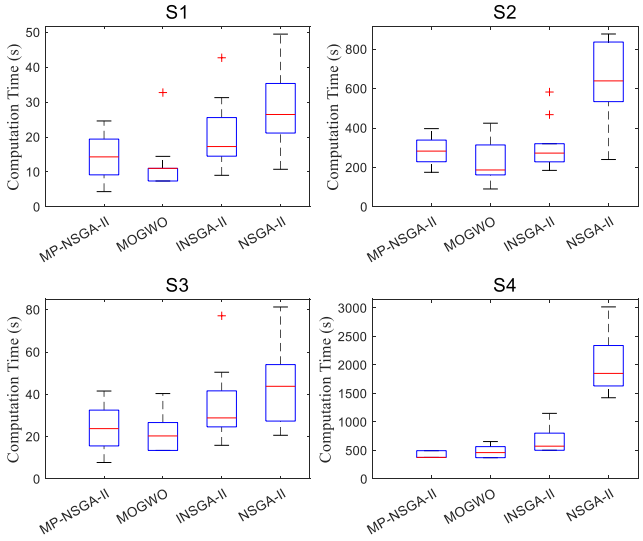


Fig. 10. Computation time for the four algorithms to obtain the complete POS.

improves as the iteration progresses. Our algorithm enhances the local search capability for potential locations of the true POS, allowing it to find solutions within the true POS with fewer iterations. Therefore, the advantage of our algorithm begins to emerge after MOGWO has gone through multiple iterations. In scenario 4, the calculation time for each iteration of the MOGWO algorithm is also greatly extended, so this advantage is more obvious.

In Fig.9, the performance of the four algorithms is compared in terms of the number of iterations required to obtain the complete POS over ten runs under four different test scenarios.

Among the four algorithms, MP-NSGA-II requires the fewest iterations to find the complete POS across all test scenarios, while NSGA-II requires the most iterations. INSGA-II ranks second in scenarios 1, 2, and 4, and third in scenario 3. The maximum number of iterations for MP-NSGA-II to find the complete POS is 5 in scenarios 1, 2, and 4, and 8 in scenario 2. MP-NSGA-II exhibits the smallest variation in the number of iterations required to obtain the complete POS across ten independent runs in all test scenarios, while NSGA-II shows the largest variation. MOGWO also demonstrates considerable variability across different runs under strong perturbation scenarios, and occasionally it takes multiple iterations to find the complete POS in weak perturbation scenarios.

Fig.10 presents box plots showing the computation time for obtaining the complete POS over 10 runs for the four algorithms. For MP-NSGA-II, the computation time ranges across 10 runs in scenarios 1-4 are as follows: 4.4-24.6s, 175.1-396.8s, 7.8-41.6s, and 378.0-496.3s, respectively. For MOGWO, the computation time ranges in scenarios 1-4 are: 7.4-32.7s, 90.6-424.3s, 13.5-40.37s, and 373.3-656.2s, respectively. In scenario 1, only one run required a relatively long time of 32.7s, while the other nine runs took no more than 14.5s. For INSGA-II, the computation time ranges in scenarios 1-4 are: 9-42.7s, 184.9-582.6s, 15.9-77.1s, and 503.2-1151s. However, in scenarios 1-3, INSGA-II has 1-2 runs with significantly longer computation times compared to the others. Excluding these outliers, the computation time ranges for INSGA-II in scenarios 1-3 are: 9-31.3s, 373-320.2s, and 15.9-41.6s. For NSGA-II, the computation time ranges in scenarios 1-4 are: 10.8-49.5s, 240.1-877.1s, 20.6-81.3s, and 1425.0-3019.3s, respectively. In terms of average computation time, MOGWO performs best in scenarios 1–3, with averages of 12.1s, 227s, and 21.6s, respectively, while MNSGA-II achieves the shortest average time in scenario 4 at 413.5s. Among the four algorithms, the average computation time of MP-NSGA-II and MOGWO is the closest across all scenarios.

Our proposed algorithm efficiently explores the entire search space through a region-based search strategy, allowing it to identify the potential locations of the true POS across different scenarios. By integrating multiple strategies and techniques, it continuously performs strong local exploration at these potential locations. This approach strikes a favorable balance between global exploration and local exploitation, ensuring that the algorithm can continuously improve its solution quality. As a result, MP-NSGA-II consistently and stably optimizes the solution across 10 runs in all four scenarios, with a relatively small number of iterations required to identify the complete POS. However, this more stable ability to continuously optimize the solution quality also increases the computation time, making each iteration of the algorithm take longer. Although the algorithm achieves the complete POS with fewer iterations, the longer computational time per iteration reduces the overall computational efficiency, thus diminishing the advantage of obtaining the complete POS in fewer iterations. During the busiest operating peak hours of the Elizabeth line, the minimum train headway is approximately

2.5 minutes (24 trains per hour). As shown in Fig.7 and Fig.8, the proposed algorithm completes its first iteration within 4-64s across all test scenarios, thereby providing decision-makers with preliminary decision support in a very short time and fully satisfying the practical application requirements. Furthermore, the algorithm can continuously optimize the obtained POS in subsequent iterations. However, finding the complete and true POS takes a relatively long time. In weak perturbation scenarios 1 and 3, the longest computation time is only 41.6s, which meets the requirements. In contrast, under strong perturbation scenarios 2 and 4, the minimum computational time required by the algorithm to find the complete and true POS reaches 175.1s, exceeding the 2.5-min requirement during peak-hour operations. Therefore, although the proposed algorithm can provide timely preliminary decision support, further improvements in computational efficiency are still required to achieve rapid convergence to the complete POS under strong perturbation and high-demand operating conditions.

The strength of NSGA-II lies in its global exploration capability, while its local search ability is relatively weak, leading to a relatively high degree of randomness in its solution process. During the early iterations, it can improve the quality of the Pareto front by finding solutions close to the true POS. However, in the later stages, further improvement requires continuously and accurately finding the remaining unexplored solutions within the true POS. This requires the algorithm to have strong local search capabilities, which is exactly what NSGA-II lacks. This limitation becomes more pronounced under strong perturbation scenarios, where the search space expands and the difficulty of locating true Pareto-optimal solutions increases significantly. Therefore, NSGA-II is prone to the situation where HV and IGD cannot be improved after multiple iterations, especially in strong perturbation scenarios. This shortcoming also leads to NSGA-II needing to undergo significantly more iterations to find the complete POS. Although NSGA-II takes less time per iteration, the significant difference in the number of iterations with the other three algorithms ultimately leads to the worst overall computational efficiency.

Building on NSGA-II, INSGA-II incorporates an external archive to preserve elite solutions and performs neighborhood search around them. This enhancement retains the strong global exploration capability of NSGA-II while significantly strengthening its local search ability. As a result, INSGA-II can continue to discover new solutions within the true POS, even in later stages of the evolutionary process, thereby consistently improving HV and IGD performance with relatively few additional iterations. This enables it to find the complete POS with relatively fewer iterations, even in strong perturbation scenarios. Although INSGA-II introduces improvements over NSGA-II, leading to enhanced algorithm performance, its optimization process still has a high degree of randomness. This randomness may explain the occasional occurrence of a relatively large number of iterations and longer computation times in scenarios 1–3.

MOGWO focuses on global exploration of the search space in the early stages of iteration, while its local exploitation capability gradually strengthens as the number of iterations increases. In weak perturbation scenarios with relatively small search spaces, MOGWO can quickly and effectively find the complete POS. However, in strong perturbation scenarios with larger search spaces, it becomes more difficult to quickly obtain the complete POS. In such cases, multiple additional iterations are typically required in the later stages to allow the algorithm's local search capability to reach a sufficient level, enabling it to discover previously missed solutions in the true POS and thereby improve HV and IGD. As a result, compared to our proposed algorithm, the computational time distribution of MOGWO tends to be more scattered. In Scenario 2, due to the relatively low computational cost per iteration, MOGWO retains a certain advantage in total runtime despite requiring a higher number of iterations. However, in Scenario 4, this per-iteration efficiency becomes less significant, and MOGWO demands even more iterations than our method. Consequently, our algorithm demonstrates an advantage in total runtime in such conditions.

Experimental results demonstrate that the improvements introduced to NSGA-II effectively balance the trade-off between global exploration and local exploitation. Compared to NSGA-II, MP-NSGA-II exhibits superior performance in solving the formulated multi-objective optimization model, often achieving higher-quality Pareto front within a shorter computation time. Additionally, MP-NSGA-II consistently improves the quality of the obtained Pareto front throughout the iterative evolution process and can identify the complete POS with fewer iterations. The algorithm also shows strong stability across 10 runs in four test scenarios. MP-NSGA-II is comparable to INSGA-II and the MOGWO, with its performance even surpassing that of INSGA-II and MOGWO in certain cases. However, this performance improvement also requires a certain computational cost, and the key improvement technology used prolongs the runtime of each evolutionary iteration.

VII. Conclusions and Future Research

This paper studies the problem of how to make rescheduled timetables exhibit a reasonable degree of robustness against uncertain perturbations. Given that rescheduled timetables often require repeated adjustments under the influence of uncertain perturbations, we propose a multi-objective optimization model with delay propagation risk and total delays as the objective function. This model aims to help dispatchers balance the trade-off between timetable robustness and its associated cost. To solve this multi-objective problem efficiently, an MP-NSGA-II algorithm is proposed by making comprehensive improvements to NSGA-II.

The effectiveness of the proposed method is validated using data from the Elizabeth Line. The experimental results show that the robust rescheduled timetable solved by our model can ensure that train operations meet the constraints under the influence of specific interval arrival deviations, thereby enabling the timetable to resist frequently occurring arrival deviations. Delay propagation risk effectively reflects the number of adjustments required by rescheduled timetables in uncertain environments, offering valuable support for

improving their robustness. The proposed MP-NSGA-II can efficiently solve the formulated multi-objective model and demonstrates clear performance improvements over NSGA-II. It also exhibits comparable performance compared to INSGA-II and MOGWO, and in some cases, even outperforms the two algorithms.

There are several promising directions in which our research can be further improved, which can be summarized into the following three aspects.

First, the proposed mathematical model can be further refined to better capture the complexity of real-world railway operations. In particular, the impacts of passenger waiting and transfers, multiple heterogeneous rail networks, diverse disruption scenarios, the cascading delay propagation relationships among trains, and multiple rescheduling strategies deserve detailed consideration in the modeling process.

Second, considerable work can also be undertaken to enhance the computational efficiency of the algorithm. Sensitivity analyses of various parameters and ablation studies should be conducted to thoroughly investigate how different parameters and algorithm components affect the POS convergence and the output results. Based on these findings, a flexible adaptive algorithm might be developed to dynamically adjust parameters and algorithmic components. Furthermore, machine learning techniques could be integrated to transfer knowledge acquired from previous solution processes to new problem instances, thereby speeding up the solution process. In addition, acceleration strategies such as parallel computing and heuristic warm-start initialization can be considered.

Third, future research should go beyond merely generating the Pareto front and delve more deeply into how to assist dispatchers in selecting the most appropriate timetable from the Pareto front.

APPENDIX

1.Constraints of station dwelling times:

$$d_{k,i} - a_{k,i} \geq d^*_{k,i} - a^*_{k,i} \quad \forall i \in S; \forall k \in T \tag{19}$$

2.Constraints of minimum section running times:

$$a_{k,i} - d_{k,i-1} \geq r_{i-1} \quad \forall i \in S \backslash \{1\}; \forall k \in T \tag{20}$$

3.Constraints of velocity limitation:

$$a_{k,i} - d_{k,i-1} \geq r^{ex}_{i-1} \quad \forall i \in S \backslash \{1\}; \forall k \in T \tag{21}$$

4.Constraints of arrival headway:

$$a_{k+1,i} - a_{k,i} \geq H_{min} \quad \forall i \in S; \forall k \in T \backslash \{m\} \tag{22}$$

5.Constraints of departure headway:

$$d_{k+1,i} - d_{k,i} \geq H_{min} \quad \forall i \in S; \forall k \in T \backslash \{m\} \tag{23}$$

6.Constraints of track capacity:

$$\begin{gathered} a_{k,i} - d_{g,i} \geq h_{min} - \left(3 - y^*_{g,k,i} - Z^*_{k,p,i} - Z^*_{g,p,i}\right) * M \\ \forall i \in S; \forall k,g \in T : k > g; \forall p \in C_i \end{gathered} \tag{24}$$

REFERENCES

[1] H. Dong, X. Liu, M. Zhou, W. Zheng, J. Xun, S. Gao, H. Song, Y. Li, and F.-Y. Wang, "Integration of Train Control and Online Rescheduling for High-Speed Railways in Case of Emergencies," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 5, pp. 1574–1582, Oct. 2022.

[2] S. Zhan, J. Xie, S. Wong, Y. Zhu, and F. Corman, "Handling uncertainty in train timetable rescheduling: A review of the literature and future research directions," *Transportation Research Part E: Logistics and Transportation Review*, vol. 183, p. 103429, Mar. 2024.

[3] D. Šemrov, R. Marsetič, M. Žura, L. Todorovski, and A. Srdic, "Reinforcement learning approach for train rescheduling on a single-track railway," *Transportation Research Part B: Methodological*, vol. 86, pp. 250–267, Apr. 2016.

[4] W. Fang, S. Yang, and X. Yao, "A Survey on Problem Models and Solution Approaches to Rescheduling in Railway Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 2997–3016, 2015.

[5] R. Hoogervorst, T. Dollevoet, G. Maróti, and D. Huisman, "Reducing Passenger Delays by Rolling Stock Rescheduling," *Transportation Science*, vol. 54, no. 3, pp. 762–784, May 2020.

[6] G. J. Polinder, T. Breugem, T. Dollevoet, and G. Maróti, "An adjustable robust optimization approach for periodic timetabling," *Transportation Research Part B: Methodological*, vol. 128, pp. 50–68, 2019.

[7] G. Cavone, M. Dotoli, N. Epicoco, and C. Seatzu, "A decision making procedure for robust train rescheduling based on mixed integer linear programming and Data Envelopment Analysis," *Applied Mathematical Modelling*, vol. 52, pp. 255–273, Dec. 2017.

[8] Z. Chen, A. D'Ariano, S. Li, M. L. Tessitore, and L. Yang, "Robust dynamic train regulation integrated with stop-skipping strategy in urban rail networks: An outer approximation based solution method," *Omega*, vol. 128, p. 103135, Oct. 2024.

[9] H. Zhang, S. Li, Y. Wang, Y. Wang, and 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," *Computers and Operations Research*, vol. 127, p. 105135, 2021.

[10] L. Meng and X. Zhou, "Robust single-track train dispatching model under a dynamic and stochastic environment: A scenario-based rolling horizon solution approach," *Transportation Research Part B: Methodological*, vol. 45, no. 7, pp. 1080–1102, 2011.

[11] T. Tang, S. Chai, W. Wu, J. Yin, and A. D'Ariano, "A multi-task deep reinforcement learning approach to real-time railway train rescheduling," *Transportation Research Part E: Logistics and Transportation Review*, vol. 194, p. 103900, Feb. 2025.

[12] A. D'Ariano, D. Pacciarelli, and M. Pranzo, "A branch and bound algorithm for scheduling trains in a railway network," *European Journal of Operational Research*, vol. 183, no. 2, pp. 643–657, Dec. 2007.

[13] J. Eaton and S. Yang, "Ant Colony Optimization for Simulated Dynamic Multi-Objective Railway Junction Rescheduling," *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, vol. 18, no. 11, pp. 2980–2992, 11/ 2017.

[14] C. Ying, A. H. Chow, Y. Yan, Y.-H. Kuo, and S. Wang, "Adaptive rescheduling of rail transit services with short-turnings under disruptions via a multi-agent deep reinforcement learning approach," *Transportation Research Part B: Methodological*, vol. 188, p. 103067, Oct. 2024.

[15] K. Sato, K. Tamura, and N. Tomii, "A MIP-based timetable rescheduling formulation and algorithm minimizing further inconvenience to passengers," *Journal of Rail Transport Planning & Management*, vol. 3, no. 3, pp. 38–53, Aug. 2013.

[16] S. Zhan, S. Wong, P. Shang, Q. Peng, J. Xie, and S. Lo, "Integrated railway timetable rescheduling and dynamic passenger routing during a complete blockage," *Transportation Research Part B: Methodological*, vol. 143, pp. 86–123, Jan. 2021.

[17] S. Long, L. Meng, Y. Wang, J. Miao, X. Luan, and F. Corman, "Integrated speed modeling and traffic management to precisely model the effect and dynamics of temporary speed restrictions to high-speed railway traffic," *Transportation Research Part C: Emerging Technologies*, vol. 152, p. 104148, Jul. 2023.

[18] S. Dündar and I. Şahin, "Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways," *Transportation Research Part C: Emerging Technologies*, vol. 27, pp. 1–15, 2013.

[19] Z. Liao, "Rescheduling Out-of-Gauge Trains with Speed Restrictions and Temporal Blockades on the Opposite-Direction Track," *Mathematics*, vol. 11, no. 12, p. 2659, Jun. 2023.

[20] C. Zhang, Y. Gao, V. Cacchiani, L. Yang, and Z. Gao, "Train rescheduling for large-scale disruptions in a large-scale railway network," *Transportation Research Part B: Methodological*, vol. 174, p. 102786, Aug. 2023.

[21] A. Bettinelli, A. Santini, and D. Vigo, "A real-time conflict solution algorithm for the train rescheduling problem," *Transportation Research Part B: Methodological*, vol. 106, pp. 237–265, Dec. 2017.

[22] G. Feng, P. Xu, D. Cui, X. Dai, H. Liu, and Q. Zhang, "Multi-stage timetable rescheduling for high-speed railways: A dynamic programming approach with adaptive state generation," *Complex & Intelligent Systems*, vol. 7, no. 3, pp. 1407–1428, 2021.

[23] P. Wang, L. Ma, R. M. Goverde, and Q. Wang, "Rescheduling trains using petri nets and heuristic search," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 726–735, 2016.

[24] Z. Bai, H. Wang, L. Yang, J. Li, and H. Lu, "A Rescheduling Approach for Freight Railway considering Equity and Efficiency by an Integrated Genetic Algorithm," *Journal of Advanced Transportation*, vol. 2023, pp. 1–22, May 2023.

[25] 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," *Journal of Advanced Transportation*, vol. 2019, pp. 13–16, 2019.

[26] H. Dong, L. Ning, M. Zhou, H. Song, and W. Bai, "Deep Reinforcement Learning for Integration of Train Trajectory Optimization and Timetable Rescheduling Under Disturbances," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2024.

[27] S. Zhan, L. G. Kroon, J. Zhao, and Q. Peng, "A rolling horizon approach to the high speed train rescheduling problem in case of a partial segment blockage," *Transportation Research Part E: Logistics and Transportation Review*, vol. 95, pp. 32–61, 2016.

[28] L. Yang, X. Zhou, and Z. Gao, "Credibility-based rescheduling model in a double-track railway network: A fuzzy reliable optimization approach," *Omega (United Kingdom)*, vol. 48, pp. 75–93, 2014.

[29] X. Liu, M. Zhou, H. Dong, X. Wu, Y. Li, and F.-Y. Wang, "ADMM-based joint rescheduling method for high-speed railway timetabling and platforming in case of uncertain perturbation," *Transportation Research Part C: Emerging Technologies*, vol. 152, p. 104150, Jul. 2023.

[30] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[31] I. Rahimi, A. H. Gandomi, K. Deb, F. Chen, and M. R. Nikoo, "Scheduling by NSGA-II: Review and Bibliometric Analysis," *Processes*, vol. 10, no. 1, p. 98, Jan. 2022.

[32] H. Ma, Y. Zhang, S. Sun, T. Liu, and Y. Shan, "A comprehensive survey on NSGA-II for multi-objective optimization and applications," *Artificial Intelligence Review*, vol. 56, no. 12, pp. 15 217–15 270, Dec. 2023.

[33] X. Hu, C. Liang, D. Chang, and Y. Zhang, "Container storage space assignment problem in two terminals with the consideration of yard sharing," *Advanced Engineering Informatics*, vol. 47, p. 101224, Jan. 2021.

[34] B. Ji, H. Huang, and S. S. Yu, "An Enhanced NSGA-II for Solving Berth Allocation and Quay Crane Assignment Problem With Stochastic Arrival Times," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 459–473, Jan. 2023.

[35] W. Li, L. Zhong, L. He, and W. Guo, "Integrated scheduling in sea–rail intermodal terminals with flexible transshipment routes," *Swarm and Evolutionary Computation*, vol. 93, p. 101852, Mar. 2025.

[36] L. Zhong, W. Li, K. Gao, L. He, and Y. Zhou, "An Improved NSGAII for Integrated Container Scheduling Problems With Two Transshipment Routes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 10, pp. 14 586–14 599, Oct. 2024.

[37] C. Wang, H. Shi, and X. Zuo, "A multi-objective genetic algorithm based approach for dynamical bus vehicles scheduling under traffic congestion," *Swarm and Evolutionary Computation*, vol. 54, p. 100667, May 2020.

[38] E. H. Haro, O. Avalos, J. Gálvez, and O. Camarena, "An Integrated Process Planning and Scheduling problem solved from an adaptive multi-objective perspective," *Journal of Manufacturing Systems*, vol. 75, pp. 1–23, Aug. 2024.

[39] L. He, R. Chiong, W. Li, G. S. Budhi, and Y. Zhang, "A multiobjective evolutionary algorithm for achieving energy efficiency in production environments integrated with multiple automated guided vehicles," *Knowledge-Based Systems*, vol. 243, p. 108315, May 2022.

[40] M. Dorigo and K. Socha, "Ant colony optimization," *Handbook of Approximation Algorithms and Metaheuristics*, no. November, pp. 26–1–26–14, 2007.

[41] J. Yuan, "Stochastic modelling of train delays and delay propagation in stations," Ph.D. dissertation, Delft University of Technology, Netherlands, 2006.

[42] S. Mirjalili, S. Saremi, S. M. Mirjalili, and L. D. S. Coelho, "Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization," *Expert Systems with Applications*, vol. 47, pp. 106–119, Apr. 2016.