Timetable Rescheduling Using Skip-Stop Strategy for Sustainable Urban Rail Transit

Zhichao Cao 1,2,3, Yuqing Wang 1, Zihao Yang 1, Changjun Chen 1 and Silin Zhang 1,*

1 School of Transportation and Civil Engineering, Nantong University, Nantong 226019, China; caozhichao@bjtu.edu.cn (Z.C.); 2118320001@stmail.ntu.edu.cn (Y.W.); 2133110116@stmail.ntu.edu.cn (Z.Y.); 2133110095@stmail.ntu.edu.cn (C.C.)
2 Technology and Equipment of Rail Transit Operation and Maintenance Key Laboratory of Sichuan Province, Chengdu 610041, China
3 Guangdong-Hong Kong-Macau Joint Laboratory for Smart Cities, Shenzhen 518100, China
* Correspondence: 12114213@bjtu.edu.cn; Tel.: +86-188-10870559

Abstract: Unanticipated events inevitably occur in daily urban rail transit operations, disturbing the scheduled timetable. Despite the mild delay, the busy operation system probably tends to worsen a larger disturbance and even lead to a knock-on disruption if no rescheduling is timely carried out. We propose a bi-objective mixed-integer linear programming model (MILP) that employs the skip-stop operation strategy to eliminate unscheduled delays. This model addresses two distinct, yet interconnected objectives. Firstly, it aims to minimize the difference between the plan and the actual operation. Secondly, it strives to minimize the number of left-behind passengers. In order to resolve this MILP problem, we devised a Pareto-based genetic algorithm (GA). Based on the case study, we certify the superior effectiveness with comparisons to the whale optimization algorithm and the epsilon constraint method. The outcomes affirm that our model has the potential to reduce the total delay time of the line by 44.52% at most compared with the traditional all-stop running adjustment model. The optimal scheme saved 6.08% of the total costs based on a trade-off between operators’ interests and passenger satisfaction.

Keywords: train delay; skip-stop; mixed-integer linear programming; Pareto-based GA; train operation diagram

1. Introduction

How to warrant the feasibility of a planned timetable during daily urban rail transit (URT) operation is an important safety-related scheduling problem. From the standpoint of users, we call the plan-based timetable “punctuality.” Otherwise, we have to undergo “delay.” In daily life, the most common delays—predefined as disturbances—are derived from congestion events. For example, train doors or platform screen doors cannot close punctually before the departure of the train, especially during peak hours. Specifically, the long-term delays and even shutoff disruptions are not taken into consideration in this work.

URT is a major contributor to public transportation and plays a crucial role in passenger transportation, promoting urbanization development. URT has good advantages, such as energy efficiency, pollution-free operation, safety, and high-quality public serviceability. Recently, URT passenger flow has been rapidly increasing in the majority of cities. The train track safety headway has been compressed so as to enhance the service frequency. In order to handle these unanticipated delays, we use a skip-stop operation to minimize the knock-on delay propagation effects.

Under normal circumstances, trains are required to precisely execute the planned timetable, which achieves punctual arrivals and departures. In reality, despite trains’ buffer times, sustaining over-congestion perhaps leads to so-called minor delays. Based
on how long the interruption lasts, URT delay events can be further divided into three groups: disturbance, delay, and disruption. Disturbances occur frequently and typically last for up to 3 minutes. They are characterized by the delay of a single train, which does not affect subsequent trains. Normal operations can be restored by adjusting the train’s running times and by using the buffer times. Delays occur less frequently and last between 3 and 15 minutes. They involve multiple subsequent train services being affected. Adjustments such as compressing buffer times and speeding up are not able to recover to normal operation. In this case, measures like skipping stations or canceling certain train services have to be triggered. Disruptions are rare and typically last for more than 15 minutes or dozens of hours. Due to the URT’s real-time operations and adjustment complexity, improper/no measures on handling delay events can escalate into even worse safety-related disruptions. Hence, it is of great significance to deal with the two former (disturbance and delay) as per reasonable strategy-based approaches to avoid a more serious disruption.

A huge number of passengers crowding on the platform during rush hours usually causes delays. These delays greatly increase passengers’ waiting time. To address this problem, this paper builds a URT timetable rescheduling model in which the intermediate stops are allowed to skip to reduce these delays. The aim is to recover to normal. From the perspective of operators, the rescheduling objective is to minimize the delay times. In terms of passenger satisfaction, the goal is to reduce the number of delayed passengers. In order to simultaneously attain the dual objectives, we formulated a bi-objective mixed-integer linear programming (MILP) model. The constraints of skip-stop, capacity-based passenger load threshold and timetabling stipulation are yielded. A Pareto-based genetic algorithm (GA) is proposed. A whale optimization algorithm and an epsilon constraint method are used as comparison metrics. At the real-time operation level, the rescheduling optimization model determines which stops to skip and is able to create a new adjusted timetable announced for operators and users simultaneously. It entails a strict load capacity as the constraint so as to exactly reduce the left-behind passengers. Eventually, the model’s validation was conducted using a real case study with comparison to the traditional timetable adjustment implement. We select a set of Pareto optimized solutions that correspond to seven feasible train adjustment schemes. These schemes are labeled by the different number of skipped stations. Furthermore, we analyze these adjustment schemes in nine scenarios.

The subsequent sections of this paper are structured as follows. We review existing studies related to skip-stop strategies in URT operations in Section 2, stating the distinctive features of our study. The skip-stop operation, the problem assumptions and nomenclature are presented in Section 3. A bi-objective MILP model is proposed in Section 4 to address the problem of sudden delay of URT trains. In an endeavor to address the model, we use a Pareto-based genetic algorithm in Section 5. A case study validates both the model’s effectiveness and the algorithm’s performance in Section 6. Finally, Section 7 encapsulates our conclusions and prospective research plan.

2. Literature Review

Hereafter, we will analyze existing studies that have utilized the skip-stop strategy as a means to deal with the rescheduling problem. They provide a solid basis for our work and help in a better understanding of our study’s features.

Train skip-stop patterns are an important strategy used in URT operation. These can be divided into a regular A/B skip pattern, as detailed by Huang et al. [1], as well as an irregular flexible skip-stop pattern, which was introduced by Nesheli et al. [2]. Abdelhafiez et al. [3] used the A/B skip-stop pattern to minimize passenger journey time. Zhang et al. [4] proposed an inventive approach for optimizing skip-stop patterns on bidirectional subway lines. Their primary goal was to minimize the average passenger travel time, which they achieved by employing a genetic algorithm for the exploration of optimal solutions. Naeini et al. [5] designed a mixed-integer programming model that addresses both the stop-skipping tactics and synchronization of timetables. Their primary objective
was to optimize the overall passenger experience by maximizing the number of passengers successfully reaching their destinations while concurrently minimizing the total number of passengers’ waiting time and in-vehicle time. To solve this, they created an heuristic algorithm based on a genetic algorithm (GA). When Cao et al. [6] developed a comprehensive evaluation model for skip-stop operation strategies, they considered factors such as passenger waiting time, journey duration, and total train travel time. To solve this model, they utilized a tabu search algorithm. Li et al. [7] aimed to maximize the reduction in total passenger journey time by employing skip-stop strategies and constructing a 0-1 integer programming model. They also employed a tabu search algorithm to solve this optimization problem.

Focusing on high passenger load of a URT system in China, Gao et al. [8] considered the issues of excessive passenger crowding and time dependency. To minimize the amount of time that passengers would be delayed by the reorganization of subway lines, they presented an optimization model. Taking into account train capacity constraints, they implemented skip-stop measures to restore the train schedule under delay conditions. To solve the model, they chose an iterative algorithm. In the study conducted by Shang et al. [9], they conceptualized individual passenger transportation and train stopping patterns into a multi-commodity flow modeling framework, considering them as commodities and network structures. Subsequently, they formulated a multi-commodity flow model to seek the optimal solution in the context of network structures. Their approach considered passenger fairness, resulting in more balanced waiting times for passengers at each station, while obtaining the optimal train skip-stop patterns. Aimed at the problem of high passenger flow at transfer stations, to identify optimal planning, Pan et al. [10] put forth a coordinated multi-line train skip-stop strategy. They formulated a binary programming model and used a genetic algorithm to identify the optimal planning. In their pursuit of addressing the challenge posed by dynamic passenger flow, Chen et al. [11] employed the skip-stop strategy. They introduced a nonlinear programming model designed to minimize total train deviations, thereby offering a solution to real-time train scheduling issues on high-frequency urban rail transit lines.

Under the circumstances of sudden delays, a large number of stranded passengers may cause platform congestion. In addition, train delays tend to increase passengers’ waiting time, so it is crucial to adjust the train schedule in a timely fashion and cater to the diverse travel requirements of passengers. Altazin et al. [12] introduced a methodology to restore train service using a skip-stop strategy for disturbances on the line. They set weights for the number of skipped stations, taking into account rolling-stock constraints. Based on this, they formulated an integer linear programming model, in which the dual objectives are minimizing both recovery times and passenger waiting times. Aimed at minimizing passenger delays, Zhu et al. [13] established a MILP model that factored in a range of strategies. These strategies encompassed flexible stopping, flexible short-turning, timetable retiming, sequencing adjustments, and even potential cancellations, all designed to facilitate the rescheduling of timetables. Li et al. [14] developed a coupled state-space optimization model to minimize timetable deviations and train headways in subway lines while achieving coordinated control of station passenger flows and train arrival times. To solve this complex model, they used the model predictive control algorithm. The adjustment of train schedules will not only involve the interests of operators but also involve passenger satisfaction. To balance the relationship between them is pivotal in ensuring the smooth operation of trains. In the efforts to address platform congestion and prolonged passenger waiting times stemming from train delays, Lu et al. [15] adopted the skip-stop strategy while simultaneously managing inbound passenger flow. Ultimately, they developed a model grounded in deep reinforcement learning via a Q-network, whose goal was to minimize the comprehensive benefits including over-limit passengers on the platform, average waiting times, and the intensity of passenger flow control. With the purpose of cooperatively optimizing train operation planning and station passenger flow control, Meng et al. [16] devised a bi-objective integer nonlinear model grounded in the
utilization of skip-stop strategies. The model aimed to enhance the efficiency of train operations and diminish the number of passengers experiencing delays. Finally, they used time reconstruction and the big-M method, incorporating 0-1 variables, to linearize the nonlinear constraints and reconstruct the model into an integer linear programming model. In summary, most existing skip-stop studies for solving train delay problems only focus on either the operators’ commercial benefits or the passengers’ interests individually. A few studies propose two objective functions to simultaneously track the trade-off problem between them [17–22].

At present, approaches to solving multi-objective optimization problems can be divided into two main groups: Pareto-based algorithms and scalarization methods [23]. Pareto-based algorithms are well suited for complex problems with multiple conflicting objectives and can attain a trade-off between the objectives. This includes multi-objective metaheuristic algorithms and the epsilon constraint method. Following are some studies that have used Pareto-based algorithms. Dulebenets et al. [24] proposed an optimization model for reactive berth allocation and scheduling in the event of unexpected disruptions at marine terminals and developed a DMO algorithm for solving this. The algorithm embedded a set of problem-specific tailored hybridization techniques to address the issue of limited interactions between the individuals located on the opposite sides of the diffusion grid. It demonstrated competitive performance against the exact mixed-integer nonlinear programming method. Fathollahi-Fard et al. [25] used a fuzzy programming approach based on triangular logic and developed an efficient multi-objective metaheuristic algorithm, namely, NSGEA, to address the issue of uncertainty in blueberry harvest planning, resulting in a more robust and accurate decision-making process for blueberry farmers. This algorithm addresses the problem-solving complexity of finding optimal harvest plans based on conflicting sustainability goals. For the vehicle routing problem with a factory in a box, Pasha et al. [26] proposed a novel bi-objective optimization model that aims to minimize the total cost associated with traversing the edges of the network and the total cost associated with visiting the nodes of the network. They designed a customized multi-objective hybrid metaheuristic solution algorithm that outperformed the ε-constraint method and several well-known metaheuristics. For the two-echelon vehicle routing problem in urban logistics, Anderluh et al. [27] considered vehicle synchronization and “gray zone” customers, and proposed a large neighborhood search method embedded in an heuristic rectangle–cuboid splitting to yield a Pareto surface and achieve a trade-off between the economic, the environmental and the social objectives.

In our study, considering the real-life delay scenarios in a URT system, we took into account the restoration of train schedules and built a bi-objective MILP model using a flexible skip-stop strategy. We solve it by using Pareto-based genetic algorithm. This model was designed to minimize both the total travel time on the line and the number of passengers experiencing delays, and finally recover from initial delays caused by a single delayed train. To substantiate the effectiveness of our approach, we conducted a case study for validation.

To provide a lucid exposition of our work, we have compiled a comparative analysis in Table 1, presenting the characteristics of pertinent studies that are closely related to our research.
Table 1. Comparison of this work with previous works on the flexible skip-stop pattern.

<table>
<thead>
<tr>
<th>Publication (Chronologically)</th>
<th>Problem</th>
<th>Number of Sub-Objectives</th>
<th>Objective(s)</th>
<th>Modelling Approach</th>
<th>Solution Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang et al., [4]</td>
<td>Optimization</td>
<td>1</td>
<td>To minimize the mean passenger travel time</td>
<td>MINLP</td>
<td>Genetic algorithm</td>
</tr>
<tr>
<td>Cao et al., [6]</td>
<td>Optimization</td>
<td>1</td>
<td>To minimize the total passenger waiting time, travel time, and train operating time</td>
<td>MIP</td>
<td>A tabu search algorithm</td>
</tr>
<tr>
<td>Gao et al., [8]</td>
<td>Optimization</td>
<td>2</td>
<td>To minimize both the total travel time and the number of waiting passengers</td>
<td>MILP</td>
<td>A heuristic algorithm</td>
</tr>
<tr>
<td>Shang et al., [9]</td>
<td>Optimization</td>
<td>1</td>
<td>To minimize the overall cost of arcs chosen by all passengers</td>
<td>LP</td>
<td>Dijkstra and Bellman-Ford</td>
</tr>
<tr>
<td>Pan et al., [10]</td>
<td>Delay</td>
<td>1</td>
<td>To minimize the total passenger delay</td>
<td>BP</td>
<td>Genetic algorithm</td>
</tr>
<tr>
<td>Altazin et al., [12]</td>
<td>Disturbance</td>
<td>2</td>
<td>To minimize both recovery times and passengers’ waiting times</td>
<td>NLIP</td>
<td>Dynamic programming algorithm</td>
</tr>
<tr>
<td>Zhu et al., [13]</td>
<td>Optimization</td>
<td>1</td>
<td>To minimize the total passenger travel time</td>
<td>MIP</td>
<td>Nested genetic algorithm</td>
</tr>
<tr>
<td>Li et al., [14]</td>
<td>Disruption</td>
<td>2</td>
<td>To minimize deviations in both the timetable and headway</td>
<td>QP</td>
<td>The MPC (Model predictive control) algorithm</td>
</tr>
<tr>
<td>Meng et al., [16]</td>
<td>Optimization</td>
<td>2</td>
<td>To minimize both the train service time and the number of passengers experiencing delays</td>
<td>MINLP</td>
<td>Time reconstruction and big-M method (CPLEX)</td>
</tr>
<tr>
<td>This paper (2023)</td>
<td>Delay</td>
<td>2</td>
<td>To minimize both the total delay time and the number of passengers experiencing delays</td>
<td>MILP</td>
<td>Pareto-based genetic algorithm</td>
</tr>
</tbody>
</table>

LP: linear programming, MIP: mixed-integer programming; MILP: mixed-integer linear programming; MINLP: mixed-integer nonlinear programming; NLIP: nonlinear integer programming; BP: binary programming; QP: quadratic programming; CPLEX: CPLEX solver.

3. Problem Statement

In this study, we focus on the train delay problem so as to restore normal operations in a timely fashion. We propose a bi-objective skip-stop optimization model. The two individual objectives are to minimize the total delay time and to minimize the total number of delayed passengers.

Essentially, the primary goal is to eliminate delays and return to normal operation. We set skipped stations as decision variables, which determine whether trains skip the certain intermediate stations so as to reduce delay. Necessary constraints are proposed for the skip-stop operation, i.e., capacity-based passenger loading and timetabling, both of which ensure the safe operation of trains and the number of boarding/alighting passengers.

To provide an explicit elucidation of these concepts, we divide the rescheduling optimization problem into three subsections. First, we illustrate the skip-stop scenario. Next,
we give reasonable assumptions for modeling. Finally, we explain the key parameters used in the model, which are crucial for our optimization decisions.

3.1. Skip-Stop Strategy

The item “skip-stop” refers to the specified trains impose passing a station(s) without stopping. Figure 1 depicts three trains implementing the skip-stop strategy. They skip three intermediate stations. When the preceding train, denoted as train $i - 1$, experiences an initial delay at station $j$, this delay tends to ripple through the system, exerting an impact on the consecutive trains, $i$ and $i + 1$. In order to restore normal operation order, train $i - 1$ skips station $j + 2$, thus mitigating the delay it initially encountered. Simultaneously, train $i$ follows suit by strategically skipping station $j$, avoiding any unnecessary stops. Similarly, train $i + 1$ also adopts the skip-stop strategy, skipping station $j + 1$. Train $i + 2$ and all subsequent trains continue to operate according to the predefined timetable without any adjustments. This skip-stop solution minimizes the delays and helps URT return to its normal operation.

Figure 1. Skip-stop strategy utilization to recover a planned timetable.

The following three factors primarily reflect how skip-stopping affects passengers.

1. Skip-stopping brings about potential time-saving benefits for passengers. By implementing this strategy, not only does it eradicate the time spent on train $i$’s dwelling at the skipped station $j$ but it also mitigates the need for the extra time allocated to the processes of acceleration and deceleration.

2. Since some passengers whose boarding station is the skipped station $j$, they cannot get on the train. As a result, they have to wait for the subsequent train $i + 1$ to get on, increasing their waiting time.

3. If destination station $j$ is skipped, some passengers cannot get off in time. These passengers have two choices in knowing the skip-stop information in advance. One is to get off at the previous station $j - 1$, then they need to spend extra time waiting for train $i + 1$ to reach the end, and the other one is to get off at station $j + 1$, and then take the reverse train from it to the end [28–30].

While the skip-stop adjustment pattern does reduce onboard passengers’ total travel time, it unavoidably generates negative effects for those passengers whose departure or arrival stations fall within the skipped ones. Therefore, once a delay occurs, how to determine which stations to skip so as to recover the plan is the main problem in this study.
3.2. Assumption

In a unidirectional URT line, the physical structure is illustrated in Figure 2. Comprising $N$ stations and $N - 1$ sections, the line facilitates the travel of trains from Station 1 to Station $N$. In the event of an unexpected event at a particular station, a train encounters an initial delay. The ensuing ripple effect stemming from this delay leads to a considerable number of passengers being marooned at stations. Consequently, this initial propagation of delay results in subsequent delays for the trains that sequentially traverse the line. It is important for trains to recover the planned timetable by determining how many stations to skip and where. By employing a flexible skip-stop strategy, our approach allows trains to bypass some certain stations without stopping. This aids in promptly recovering from train delays and facilitating adjustments to the train schedule. Our optimization objectives encompass two primary facets: first, the reduction in the cumulative count of passengers experiencing delays, aimed at enhancing passenger satisfaction; second, the minimization of the total delay time experienced throughout the operation of the line. To tackle these objectives effectively, we have formulated a bi-objective MILP model, which has been designed with the explicit aim of generating a new timetable solution that addresses both passenger satisfaction and operator interests.

Figure 2. Schematic diagram of physical structure of rail transit line.

We propose the following assumptions to facilitate URT rescheduling modeling.

4. Disruption events such as rail accidents, train breakdown, track damage, are not taken into account in this paper.

5. Trains can adopt a flexible skip-stop operation to reschedule the planned timetable. However, the first and last stations are forbidden to be skipped. In addition, the train running time is fixed.

6. Due to the unexpected train delay, it is imperative to account for the scenario in which trains $1, 2, \ldots, k - 1$ have already commenced their journeys from station $j$ prior to the occurrence of the delay for train $k$ at the same station. Our model specifies that trains $1, 2, \ldots, k - 1$ cannot change their timetables midway.

7. Train $k$ experiences an initial delay at station $j$, and the subsequent trains $k + 1, k + 2, \ldots, n$ need to wait at station $j - 1$ and then operate based on the adjusted timetable.

8. Utilizing the known passenger arrival rates, we conduct calculations to determine the passenger demand origin–destination (OD) for each station. Given the consistency of OD pairs during peak hours, it is reasonable to use these data as inputs for passenger demand.

3.3. Nomenclature

The train scheduling operation framework is composed of train set $M = \{1, 2, \ldots, i \ldots, m\}$ and station set $N = \{1, 2, \ldots, j \ldots, n\}$. Once we know which train is delayed at which station, and at what time it begins and how long it lasts, the skip-stop adjustment is triggered. Table 2 presents a comprehensive compilation of the notations and parameters employed in the context of this study.
Table 2. Comprehensive compilation of notations, variables, and their descriptive explanations.

<table>
<thead>
<tr>
<th>Sets</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>Set of trains, $M = {1,2,\ldots,i,\ldots,m}$</td>
</tr>
<tr>
<td>$N$</td>
<td>Set of stations, $N = {1,2,\ldots,j,\ldots,n}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s$</td>
<td>Sequence number of delayed train</td>
</tr>
<tr>
<td>$u$</td>
<td>Sequence number of the station at which the train encounters a delay</td>
</tr>
<tr>
<td>$t_0$</td>
<td>Time when the initial delay occurred</td>
</tr>
<tr>
<td>$O$</td>
<td>Duration of the delay</td>
</tr>
<tr>
<td>$D_{i,j}$</td>
<td>Planned station halt duration for train $i$ at station $j$</td>
</tr>
<tr>
<td>$R_{i,j}$</td>
<td>Anticipated travel duration for train $i$ between station $j$ and station $j+1$</td>
</tr>
<tr>
<td>$a_{i,j}/d_{i,j}$</td>
<td>Planned time for train $i$ to arrive/depart at station $j$</td>
</tr>
<tr>
<td>$H_1$</td>
<td>Minimum headway of trains</td>
</tr>
<tr>
<td>$H_2$</td>
<td>Planned headway</td>
</tr>
<tr>
<td>$L$</td>
<td>Minimum arrival and departure interval for a train</td>
</tr>
<tr>
<td>$C_{\text{max}}$</td>
<td>Train’s maximum capacity</td>
</tr>
<tr>
<td>$X$</td>
<td>Allowable maximum number of skip-stop instances for all trains</td>
</tr>
<tr>
<td>$\tau_1/\tau_2$</td>
<td>Additional time for train acceleration/deceleration (unit: sec)</td>
</tr>
<tr>
<td>$\lambda_j$</td>
<td>Passenger arrival rate of station $j$ (unit: pax/sec)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_{i,j}'/d_{i,j}'$</td>
<td>Actual time at which train $i$ arrive at/depart from station $j$</td>
</tr>
<tr>
<td>$\alpha_{i,j,k}$</td>
<td>Within the interim span between the departure of train $i-1$ from station $j$ and the subsequent arrival of train $i$ at the same station, the count of passengers reaching station $j$ with an intent to board train $i$ for their travel to station $k$</td>
</tr>
<tr>
<td>$\beta_{i,j,k}$</td>
<td>Number of passengers granted permission to get on train $i$ for travel between station $j$ and station $k$</td>
</tr>
<tr>
<td>$B_{i,j}$</td>
<td>Number of passengers getting on train $i$ at station $j$</td>
</tr>
<tr>
<td>$E_{i,j}$</td>
<td>Number of passengers getting off train $i$ from station $j$</td>
</tr>
<tr>
<td>$P_{i,j}$</td>
<td>Number of passengers in train $i$ upon its departure from station $j$</td>
</tr>
<tr>
<td>$A_{j,k}$</td>
<td>Total number of passengers reaching station $j$, all of whom express an intention to travel to station $k$</td>
</tr>
<tr>
<td>$L_{j,k}$</td>
<td>Number of passengers who have successfully getting on trains at station $j$ and will travel to station $k$</td>
</tr>
<tr>
<td>$\delta_{i,j}$</td>
<td>A binary variable whose value is set to 1 in the event of train $i$ encountering a delay, otherwise 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{i,j}$</td>
<td>A binary variable which holds a value of 1 when train $i$ stops at station $j$, otherwise 0</td>
</tr>
</tbody>
</table>

4. Formulation of the Model

In order to establish a logical framework and facilitate an explicit approach, we divide this section into two subsections. First, two objective functions are proposed. Second, necessary constraints are proposed with regard to skip-stop, load capacity, and timetabling.
4.1. Objective Functions

Two separate objective functions are presented within this section, offering dual perspectives from both operators and passengers:

\[
\min \left\{ \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \cdot \delta_{ij} \cdot (a_{ij}' - a_{ij}) + \sum_{j=1}^{n} x_{ij} \cdot \delta_{ij} \cdot (d_{ij}' - d_{ij}) \right\} \\
\min \sum_{j=1}^{n} \sum_{k=j+1}^{n} (A_{j,k} - L_{j,k}) \quad (1)
\]

where objective function (1) is formulated to minimize the deviation between actual and planned arrival/departure times of rolling stock at/from all stations. Objective function (2) is geared towards minimizing the number of passengers left behind so as to reduce their frustration.

4.2. Constraints

We here state some crucial constraints, including the train skip-stop constraints, capacity-based passenger loading constraints and timetabling constraints, respectively.

4.2.1. Skip-Stop Constraint

The four constraints related to the skip-stop operation are listed below, specifically addressing consecutive skip operations.

\[
\sum_{i \in M} \sum_{j \in N} x_{ij} \leq X \quad (3)
\]

\[
x_{i,1} + x_{i,n} = 2 \quad \forall i \in M \quad (4)
\]

\[
x_{ij} + x_{i,j+1} \geq 1 \quad \forall i \in M, j \in N \quad (5)
\]

\[
x_{ij} + x_{i+1,j} \geq 1 \quad \forall i \in M, j \in N \quad (6)
\]

Constraint (3) restricts the count of skipped stations for all trains. Constraint (4) serves as an assurance, ensuring that each train makes scheduled stops at both the original and final stations. Constraint (5) regulates that a train is precluded from skipping two consecutive stations. Constraint (6) stipulates that a station being skipped by two successive trains is prohibited.

4.2.2. Capacity-Based Passenger Loading Constraint

Subject to train capacity for loading, we divide the boarding passengers, alighting passengers, and left-behind passengers for each train at every station. The detailed calculations are listed as follows.

\[
A_{j,k} = \sum_{i=1}^{m} \alpha_{i,j,k} \quad \forall j \in N, k \in N, \quad (7)
\]

\[
\alpha_{i,j,k} = \lambda_j \cdot (a_{i+1,j} - d_{i,j}) \quad \forall i \in M, j \in N, k \in N, \quad (8)
\]

\[
L_{j,k} = \sum_{i=1}^{m} \beta_{i,j,k} \quad \forall j \in N, k \in N, \quad (9)
\]
\begin{align*}
\{ \beta_{i,j,k} > 0, x_{i,j} \cdot x_{i,k} = 1 \} \\
\{ \beta_{i,j,k} = 0, x_{i,j} \cdot x_{i,k} = 0 \} \forall i \in M, j \in N, k \in N, \tag{10}
\end{align*}

\begin{equation}
B_{i,j} = \sum_{k=j+1}^{n} \beta_{i,j,k} \forall i \in M, j \in N, \tag{11}
\end{equation}

\begin{align*}
\{ E_{i,k} = 0, k = 1 \} \\
\{ E_{i,k} = \sum_{j=1}^{k-1} \beta_{i,j,k}, k \neq 1 \} \forall i \in M, k \in N, \tag{12}
\end{align*}

\begin{align*}
\{ P_{i,j} = B_{i,j}, j = 1 \} \\
P_{i,j} = 0, j = m \quad & \forall i \in M, j \in N, k \in N, \tag{13} \\
P_{i,j} = P_{i,j-1} + B_{i,j} - E_{i,j}, j \neq 1, j \neq m \\
P_{i,j} \leq C_{\text{max}} \forall i \in M, j \in N, \tag{14}
\end{align*}

\begin{equation}
\sum_{i=1}^{m} B_{i,j} = \sum_{i=1}^{m} \sum_{k=j+1}^{n} \beta_{i,j,k} \forall j \in N, \tag{15}
\end{equation}

Constraint (7) calculates the quantity of passengers \( A_{i,j,k} \) with an intention to commence their trip from station \( j \) to station \( k \). Here, the calculation of \( \alpha_{i,j,k} \) is achieved by multiplying the passenger arrival rate at station \( j \) denoted as \( \lambda_j \) by the specified time interval, as delineated in constraint (8). Constraint (9) calculates the quantity of passengers, represented as \( L_{j,k} \), who are both eligible to access the platform at station \( j \) and eager to commence their travel to station \( k \). Constraint (10) elucidates the correlation between the quantity of passengers \( \beta_{i,j,k} \) who board the train and the skip-stop strategy applied during the train's operation. Constraint (11) calculates the quantity of passengers \( B_{i,j} \) who have clearance to enter station \( j \) and embark on train \( i \). Constraint (12) calculates the quantity of passengers \( E_{i,k} \) alighting from train \( i \) upon its arrival at station \( j \). Constraint (13) calculates the quantity of passengers \( P_{i,j} \) present on board train \( i \) as it commences its travel from station \( j \). Constraint (14) imposes a limitation ensuring that the quantity of passengers on a train does not surpass the designated maximum capacity \( C_{\text{max}} \). Constraint (15) guarantees that the total train capacity over the operational period satisfies the overall travel demand from passengers, meaning all passengers are able to board the trains.

**4.2.3. Timetabling Constraint**

The timetabling constraints associated with rolling stock arrivals/departures at/from each station yielding to the safety tracking headway can be expressed by:

\begin{equation}
d_{i,j} = a_{i,j} + x_{i,j} \cdot D_{i,j} \forall i \in M, j \in N, \tag{16}
\end{equation}

\begin{align*}
a_{i,j+1} = d_{i,j} + R_{i,j} - x_{i,j} \cdot \tau_1 - x_{i,j+1} \cdot \tau_2 \forall i \in M, j \in N, \tag{17} \\
a_{i,j+1} = d_{i,j} + R_{i,j} \forall i \in M, j \in N, \tag{18}
\end{align*}

\begin{equation}
d'_{i,j} \geq d_{i,j} \forall i \in M, j \in N, \tag{19}
\end{equation}

\begin{equation}
d'_{i,j} - a'_{i,j} \geq D_{i,j} \forall i \in M, j \in N, \tag{20}
\end{equation}

\begin{equation}
a'_{i,j+1} - d'_{i,j} \geq R_{i,j} \forall i \in M, j \in N, \tag{21}
\end{equation}
where constraint (16) calculates the planned departure time $d_{ij}$ for train $i$ from station $j$. Constraint (17) calculates the planned arrival time $a_{ij+1}$ for train $i$ as it arrives at station $j+1$. To streamline the formulation and minimize the influence of the impact of $\tau_1$ and $\tau_2$ on the result, we simplify them in constraint (18). Constraint (19) ensures that the actual departure time $d_{ij}$ for train $i$ from station $j$ adheres to the planned scheduled departure time $d_{ij}$ or exceeds it. Constraint (20) serves to guarantee that the actual dwelling time for train $i$ at station $j$ equals or surpasses the planned dwelling time $D_{ij}$. Constraint (21) assumes a critical role by safeguarding that the actual travel time between stations for train $i$ remains at or surpasses the planned travel time $R_{ij}$. Constraint (22) ensures the safety headway between adjacent trains. Constraint (23) limits the time intervals for arrivals and departures of the adjacent trains.

In summary, the two conflicting objectives are essential for optimizing the URT system jointly. From the operator’s perspective, minimizing the total route delay time is of paramount importance. This objective emphasizes the operation efficiency, adherence to schedules, and the plan’s reliability. On the other hand, from the perspective of passenger satisfaction, minimizing the total number of delayed passengers takes precedence. The second objective function places the passengers’ experience at the forefront, striving to reduce the number of delayed users. It aims to alleviate the frustration and inconvenience derived from unexpected delays.

The inherent multi-objective conflict means that a single-objective optimization cannot accurately capture the intricate relationship between operator efficiency and passenger satisfaction. To seek the trade-off, we employ the multi-objective optimization method. This allows us to explore a series of solutions along the Pareto fronts, in which improving one objective comes at the expense of the other. By generating this trade-off curve, operators can make informed decisions based on their priorities.

5. Solution Algorithm

We have 3 intractable decision-making problems to handle: (1) the number of skipped stations, (2) the location of skip-stop stations, and (3) timetabling adjustment, respectively. It is hard to resolve this integration problem straightforwardly. The complexity of the model motivates us to divide the solution approach into three sub-algorithms to achieve a modular and better-structured solution. The main algorithm 1 (Section 5.2.1) serves as the core of our approach. It is apt for balancing the two objective functions by iteratively determining the number of skipped stations and their locations. Sub-algorithm 2 (Section 5.2.2) primarily provides data input and calculates objective functions. It outputs optimization decisions. Sub-algorithm 3 (Section 5.2.3) is the practical execution of the Pareto-based genetic algorithm (GA) for optimization until the terminal condition. Essentially, these three sub-algorithms work together, with each contributing to the overall optimization process.

5.1. Algorithm Applicability

With the purpose of solving the proposed model, we developed a Pareto-based heuristic algorithm to analyze a series of feasible Pareto solutions yielding to an acceptable time window. The algorithm framework is composed of three modules: main algorithm 1 (Section 5.2.1), sub-algorithm 2 (Section 5.2.2), and sub-algorithm 3 (Section 5.2.3). First, main algorithm 1 is to attain the trade-off between the two objective functions by iterating the number of skipped stations and where they are. Second, sub-algorithm 2 calculates the objective functions and processes the constraints. Last, sub-algorithm 3 executes the Pareto-based genetic algorithm (GA) for optimization until the terminal condition. In
other words, main algorithm 1 invokes sub-algorithm 3 to perform the optimization process to find the optimized train schemes. Sub-algorithm 3 needs to invoke sub-algorithm 2 to obtain the objective function values to evaluate the fitness and iteratively search for the optimized parametric solution according to the fitness value.

Since we are confronting a bi-objective optimization problem and these two objective functions are mutually conflicting, it is necessary to strike a balance between these two objectives. The Pareto-based GA is capable of illustrating the trade-off and compromise between different objectives [31–32]. Additionally, the Pareto-based GA possesses a strong global search capability, enabling us to discover a set of non-dominated solutions, presenting multiple feasible optimization schemes for selection, with flexibility [33–35]. Therefore, we opt for utilizing a Pareto-based heuristic algorithm for optimization.

5.2. Algorithm Framework

We present the algorithm framework divided into three parts: main algorithm 1, expounded upon in Section 5.2.1; sub-algorithm 2, comprehensively discussed in Section 5.2.2; and finally, sub-algorithm 3, meticulously examined in Section 5.2.3. The complete logic is exhibited in Figure 3.

![Algorithm flowchart](image-url)
5.2.1. Main Algorithm 1

Main algorithm 1 maintains the execution flow of the whole program. It includes the following three steps: (i) definition of parameters and variables, (ii) inputting train operation information and station information, and (iii) invoking sub-algorithms. The procedures of the main algorithm are presented in detail below.

Step 1: Parameter setting.
The number of trains and stations, planned headway, minimum headway, start time, dwelling time, running time, the initial delayed train’s sequence number, station’s sequence number where the initial delay occurred, duration of the delay, and passenger arrival rates are all included in the parameter set.

Step 2: OD passenger flow calculation.
Input station information, then calculate and generate station demand data per each 5 minutes.

Step 3: Determine the range of decision variables.
The binary decision variables determine whether the train skips a station or not. We stipulate the search space by defining a range with a minimum value $x_{limit}_{min}$ and a maximum value $x_{limit}_{max}$.

Step 4: Invoke sub-algorithm 3.

Step 5: Use the output of step 8 in sub-algorithm 3 as the input for the current step. Normalize multiple sets of objective function values on the Pareto front and compare them using a linear weighting method to select the optimal scheme.

Step 6: Based on the objective function values of the optimized scheme, extract the corresponding parameter values $x$.

Step 7: Output optimized scheme.
Obtain the two objective function values and the actual train operation timetable.

5.2.2. Sub-Algorithm 2

We present the detailed procedures of calculating the objective functions here, following step 1 in sub-algorithm 3. Accordingly, we continue GA-based iteration based on the objective function values (derived from Sub-algorithm 2, here) in the following sub-algorithm 3.

Step 1: Convert the input $x$ into a matrix and modify it to get $X_1$.
$X_1$ is used to represent train scheduling decisions.

Step 2: Establish constraints (4)-(6) for train skip-stop operation to reconstruct $X_1$.

Step 3: For each train, calculate the anticipated arrival and departure timings under normal and actual conditions.

Step 4: Calculate the total delay time of the line based on Eq. (1).

Step 5: Calculate passenger flow information at each platform.

Step 6: Calculate the total number of delayed passengers based on Eq. (2).

Step 7: Design a penalty mechanism to eliminate schemes with excessive skip-stop instances.

Step 8: Return $yy$.
Obtain the evaluation metric $yy: [y_1, y_2]$ for schemes filtered through the penalty mechanism. $y_1$ represents the total number of delayed passengers; $y_2$ represents the total delay time.

5.2.3. Sub-Algorithm 3

In order to minimize the objective functions, we use the Pareto-based GA (that corresponds to step 4 in Main algorithm 1) to find a set of Pareto fronts. Here, “Pareto fronts” refer to several feasible adjustment schemes. Here are the steps:

Step 1: Initialize population $ori_x$.

$ori_x$ is a size of 2000 * 60 random matrix, with every element has been generated through a process of randomization, conforming to a uniform distribution that spans the
numerical range from 0 to 1. Each row represents an individual \( x \). Each dimension of each individual corresponds to a stay of solving parameters \( x_{i,j} \).

**Step 2:** Fitness evaluation.
Utilize the Pareto ranking method. Specifically, begin by designating all schemes as non-dominated, then establish dominance relationships and assign level sequences by comparing their objective function values until all schemes are allocated to distinct levels. Furthermore, construct the Pareto front in order of level ranks. The fitness value for each scheme is calculated based on its level rank, with lower-level ranks receiving higher fitness values.

**Step 3:** Selection.
Select schemes from the Pareto front according to their fitness values using the roulette wheel method.

**Step 4:** Crossover.
Perform crossover operations with a probability of 0.8 on the selected schemes to generate new schemes. The specific operation involves randomly selecting genes on individual \( x \) for pairwise exchange.

**Step 5:** Mutation.
Execute mutation operations that involve swapping schemes with a probability of 0.05.

**Step 6:** Population regeneration.
The parent individual and the new individual obtained by crossover and mutation are combined into the next-generation population.

**Step 7:** Determine whether the termination condition is met.
According to the value of \( \text{StallGenLimit} \), when there is no significant change in the values of objective functions within 100 successive generations, the algorithm terminates; otherwise, it continues to perform **step 2** to **6**.

**Step 8:** Output the Pareto front.
The final returned results include \( \text{group.FinalX} \) and \( \text{group.FinalValu} \).
\( \text{group.FinalX} \) is a matrix containing multiple non-dominated solutions, each row representing a non-dominated solution and each column corresponding to the value of an unknown parameter. More generally, \( \text{group.FinalX} \) is the final individual or optimized individual of the chromosome. It represents some of the optimized schemes outputted by the Pareto-based GA. \( \text{group.FinalValu} \) is a matrix where each row is a set of objective function values corresponding to a solution vector. In essence, \( \text{group.FinalValu} \) is the final fitness value for each individual in the Pareto-based GA.

6. Case Study
6.1. Data and Parameter Set

A URT line case including eight stations is testified in this section, depicted in Figure 4. On this single-direction line, 10 trains operate yielding minimum safety headway \( H_1 \), 100s, which is a typical value to maintain a safe distance between trains and prevent rear-end collision, as found in the research of Cui et al. [36]. In the planning level, the scheduled headway \( H_2 \), of 180 seconds is derived from operational data based on the average headway values of major urban rail transit lines, such as in the case studies conducted by Li et al. [14]. The maximum capacity for each train, 1400 passengers, is in line with current high-capacity trains used in metro systems worldwide, as reported by Yuan et al. [37].

![Figure 4. Time allocation of each train on the line.](image-url)
The first train departs at 8:00:00 AM, and the last train (the 10th train) ends its operation at 8:43:06 AM. The detailed train timetable is shown in Table 3. Figure 5 illustrates the train operation diagram plan. The time interval is discrete and the time unit is seconds. To be clear, we set the first train’s departure time (8:00:00) to time point 0. Accordingly, the second departure time, 8:03:00, is classified as the 180th second. The definition of the third train’s departure time, 8:06:00, is the 360th second. The subsequent departure time also adheres to this discrete criterion.

Figure 5. Timetable of the planned scheme.

Allowing for the disturbance activities, we create a hypothesis scenario in which the second train encounters a disturbed event leading to an initial delay at station $S_2$. The delay time is 4 minutes. Thus, the actual departure time is 8:09:25, although the planned departure time is 8:05:25. The passenger arrival rates per five minutes for each of the seven stations are shown in Table 4. As per constraint (8), the accumulated passenger flow between adjacent trains’ arrival time intervals at the stations can be calculated (Table 5).

Table 3. Planned train arrival and departure schedule.

<table>
<thead>
<tr>
<th>Train</th>
<th>Station</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
<th>$S_5$</th>
<th>$S_6$</th>
<th>$S_7$</th>
<th>$S_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train_1</td>
<td>—</td>
<td>$a_{1,2}$</td>
<td>8:01:55</td>
<td>$a_{1,3}$</td>
<td>8:04:45</td>
<td>$a_{1,4}$</td>
<td>8:06:40</td>
<td>$a_{1,5}$</td>
<td>8:09:25</td>
</tr>
<tr>
<td></td>
<td>$d_{1,1}$</td>
<td>8:00:00</td>
<td>0</td>
<td>$d_{1,2}$</td>
<td>8:02:25</td>
<td>$d_{1,3}$</td>
<td>8:05:20</td>
<td>$d_{1,4}$</td>
<td>8:07:10</td>
</tr>
<tr>
<td>Train_2</td>
<td>—</td>
<td>$a_{2,2}$</td>
<td>8:04:55</td>
<td>$a_{2,3}$</td>
<td>8:07:45</td>
<td>$a_{2,4}$</td>
<td>8:09:40</td>
<td>$a_{2,5}$</td>
<td>8:12:25</td>
</tr>
<tr>
<td></td>
<td>$d_{2,1}$</td>
<td>8:03:00</td>
<td>0</td>
<td>$d_{2,2}$</td>
<td>8:05:25</td>
<td>$d_{2,3}$</td>
<td>8:08:20</td>
<td>$d_{2,4}$</td>
<td>8:10:10</td>
</tr>
<tr>
<td>Train_3</td>
<td>—</td>
<td>$a_{3,1}$</td>
<td>8:07:55</td>
<td>$a_{3,2}$</td>
<td>8:10:45</td>
<td>$a_{3,3}$</td>
<td>8:12:40</td>
<td>$a_{3,4}$</td>
<td>8:15:25</td>
</tr>
<tr>
<td></td>
<td>$d_{3,1}$</td>
<td>8:06:00</td>
<td>0</td>
<td>$d_{3,2}$</td>
<td>8:08:25</td>
<td>$d_{3,3}$</td>
<td>8:11:20</td>
<td>$d_{3,4}$</td>
<td>8:13:10</td>
</tr>
<tr>
<td>Train_4</td>
<td>—</td>
<td>$a_{4,1}$</td>
<td>8:10:55</td>
<td>$a_{4,2}$</td>
<td>8:13:45</td>
<td>$a_{4,3}$</td>
<td>8:15:40</td>
<td>$a_{4,4}$</td>
<td>8:18:25</td>
</tr>
</tbody>
</table>
6.2. Computation Results

The Pareto-based GA and the WOA used here were programmed by MATLAB 2020R on a personal computer with an Intel Core 11th i5 CPU @ 3.10 GHz, and 16.00 G RAM. We get a set of Pareto solutions as shown in Figure 6, where each Pareto optimized solution corresponds to a feasible rescheduling scheme. The Pareto solutions give rise to various objective function values (Figure 6), depending on how many stations can be skipped.

As validated in Figure 6, the values of the two objective functions show a reverse change trend. The delay time can be decreased by no more than 1400 seconds and the

---

Table 4. Passenger arrival rate.

<table>
<thead>
<tr>
<th>Station</th>
<th>Passenger Arrival Rate (pax/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>1.036</td>
</tr>
<tr>
<td>$S_2$</td>
<td>0.953</td>
</tr>
<tr>
<td>$S_3$</td>
<td>0.821</td>
</tr>
<tr>
<td>$S_4$</td>
<td>0.593</td>
</tr>
<tr>
<td>$S_5$</td>
<td>0.421</td>
</tr>
<tr>
<td>$S_6$</td>
<td>0.234</td>
</tr>
<tr>
<td>$S_7$</td>
<td>0.150</td>
</tr>
</tbody>
</table>

Table 5. Passenger flow of each station.

<table>
<thead>
<tr>
<th>Destination</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
<th>$S_5$</th>
<th>$S_6$</th>
<th>$S_7$</th>
<th>$S_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>17</td>
<td>14</td>
<td>25</td>
<td>25</td>
<td>21</td>
<td>19</td>
<td>24</td>
</tr>
<tr>
<td>$S_2$</td>
<td></td>
<td>22</td>
<td>27</td>
<td>21</td>
<td>31</td>
<td>27</td>
<td>15</td>
</tr>
<tr>
<td>$S_3$</td>
<td></td>
<td></td>
<td>18</td>
<td>31</td>
<td>19</td>
<td>15</td>
<td>21</td>
</tr>
<tr>
<td>$S_4$</td>
<td></td>
<td></td>
<td></td>
<td>22</td>
<td>32</td>
<td>23</td>
<td>12</td>
</tr>
<tr>
<td>$S_5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>27</td>
<td>12</td>
<td>22</td>
</tr>
<tr>
<td>$S_6$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14</td>
<td>20</td>
</tr>
<tr>
<td>$S_7$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>21</td>
</tr>
</tbody>
</table>

**6.2. Computation Results**

The Pareto-based GA and the WOA used here were programmed by MATLAB 2020R on a personal computer with an Intel Core 11th i5 CPU @ 3.10 GHz, and 16.00 G RAM. We get a set of Pareto solutions as shown in Figure 6, where each Pareto optimized solution corresponds to a feasible rescheduling scheme. The Pareto solutions give rise to various objective function values (Figure 6), depending on how many stations can be skipped.

As validated in Figure 6, the values of the two objective functions show a reverse change trend. The delay time can be decreased by no more than 1400 seconds and the
number of delayed passengers will rise by around 950 pax as the total number of skip-stop instances for all trains increases. According to the findings, the line’s total delay time decreases as the trains skip more stations and spend less time at each one. However, skipping stations results in some passengers who are not able to get on and off at the planned stations, which has led to an increasing number of delayed passengers. To sum up, restoring the train operation through the skip-stop strategy also sacrifices the traveling time of some passengers, especially those who had planned to get on and off at the skipped stations.

Figure 6. A set of Pareto fronts.

We select seven Pareto fronts from Figure 6 for further comparative analysis. The number of skipped stations in each of the seven train operating adjustment schemes is 0, 1, 2, 3, 4, 5, and 6, correspondingly. In Table 6, it is calculated how many passengers were delayed overall and how long the delay lasted on the line for each of these seven schemes.

To delve deeper into examining the correlation between operators’ interests and passenger satisfaction, we introduced the penalty coefficient $P_1$ (unit: $/second) and $P_2$ (unit: $/pax). We have $T$ to represent the total delay time and $P$ to represent the delayed passengers’ total number. We multiply them by the corresponding penalty coefficient $P_1$ and $P_2$, respectively, to obtain Operator Cost (unit: $) and Passenger Costs (unit: $) in Eq. (24) and Eq. (25). Finally, in Eq. (26), we add the two together to get the Overall Costs (unit: $).

\[
\text{Operator Costs} = P_1 \cdot T,
\]

\[
\text{Passenger Costs} = P_2 \cdot P,
\]

\[
\text{Overall Costs} = \text{Operator Costs} + \text{Passenger Costs},
\]

With set $P_1 = P_2 = 1$, we can calculate the Passenger Costs, Operator Costs, and Overall Costs for the seven schemes. By comparing Overall Costs, we determine the saving percentage. The specific values are also presented in Table 6.

We observe that the more stations the train skips (up to a maximum of six stations of 10 trains), the more delayed passengers and the shorter the total delay time. To compare Scheme 1 as the basic scheme with the other six schemes, we calculate the proportion of the increase or decrease of the two objective function values in the other six schemes compared with those in Scheme 1, as exhibited in Table 6. Scheme 1 only restores train opera-
tion by adjusting the departure interval of trains, which is the traditional all-stop adjustment strategy. In contrast to Scheme 1, Schemes 2, 3, 4, 5, 6, and 7 adjust the train departure intervals and simultaneously implement the skip-stop strategy to adjust train operations. In contrast, Scheme 1 has the fewest overall passengers who are delayed and the longest aggregate delay duration of the line, whereas Scheme 7 is the exact reverse. When we introduce penalty coefficients and take them to the same value, it means that we take an equal trade-off between the penalty levels for both objective functions. As a consequence, Scheme 5 is the optimized scheme when compared to Scheme 1, saving around 6.08% of the overall cost.

Table 6. Model optimization results.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Number of the Skipped Stations of all Trains</th>
<th>Total Number of Delayed Passengers (Unit: Pax)</th>
<th>Total Delay Time of the Line (Unit: Second)</th>
<th>Overall Costs (Unit: $)</th>
<th>Overall Costs Saving Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(basic)</td>
<td>0</td>
<td>633</td>
<td>2080</td>
<td>2713</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>788 (↑ 24.49%)</td>
<td>1832 (↓ 11.92%)</td>
<td>2620</td>
<td>3.43%</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1064 (↑ 68.09%)</td>
<td>1532 (↓ 26.35%)</td>
<td>2596</td>
<td>4.31%</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1163 (↑ 83.73%)</td>
<td>1407 (↓ 32.36%)</td>
<td>2570</td>
<td>5.27%</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1242 (↑ 96.21%)</td>
<td>1306 (↓ 37.21%)</td>
<td>2548 (minimum)</td>
<td>6.08%</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>1395 (↑ 120.38%)</td>
<td>1203 (↓ 42.16%)</td>
<td>2598</td>
<td>4.24%</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>1543 (↑ 143.76%)</td>
<td>1154 (↓ 44.52%)</td>
<td>2697</td>
<td>0.59%</td>
</tr>
</tbody>
</table>

In order to further analyze the characteristics from Scheme 1 to Scheme 7, we present the specific details of each timetable schemes, which correspond to Figures 7, 8, 9, 10, 11, 12 and 13, respectively. They explicitly depict the trajectory of the planned scheme, trajectory of the adjusted scheme, all the skipped stations by the trains, and the adjusted departure intervals between trains.

As can be seen from Figure 7, in Scheme 1, the trains restore to the planned operating state only by adjusting the departure intervals without skip-stopping. Specifically, after the second train delayed at the second station, we need to shorten the departure intervals between the third and fourth trains, the fourth and fifth trains, and the fifth and sixth trains to 135 seconds. Additionally, we need to shorten the departure interval between the sixth and seventh trains to 140 seconds and between the seventh and eighth trains to 165 seconds. By making these adjustments, both the ninth and tenth trains will be restored to the planned operating state.

We can observe from Figure 8 that in Scheme 2, only the second train skips the third station. Additionally, by shortening the departure interval between the third and fourth trains to 130 seconds, the interval between the fourth and fifth trains, the fifth and sixth trains, and the sixth and seventh trains to 135 seconds, and the interval between the seventh and eighth trains to 175 seconds, subsequent trains, including the ninth train, can be restored to the planned operating state.
There are a total of two skip-stop incidents for all trains in Scheme 3, as shown in Figure 9: the second train skips the third station, and the third train skips the second station. Additionally, we shorten the departure interval between the third and fourth trains to 100 seconds, the interval between the fourth and fifth trains and fifth and sixth trains to 135 seconds, and the interval between the sixth and seventh trains to 160 seconds. By doing so, starting from the eighth train, all subsequent trains can be restored to the planned operating state.
Figure 9. Timetable of Scheme 3.

In Figure 10, we can see that there were three skip-stop instances for all trains in Scheme 4: the second train skips the third and fifth stations, and the third train skips the second station. The adjustment of departure intervals between trains is the same as in Scheme 3. Starting from the eighth train, all subsequent trains have been restored to the planned operating state.

Figure 10. Timetable of Scheme 4.

In Figure 11, Scheme 5, the second train skips the third and fifth stations, and the third train skips the second and fourth stations, making a total of four skip-stop instances.
The adjustment of train departure interval is also the same as Scheme 3 and Scheme 5, and resumes its planned operation from the eighth train.

Figure 11. Timetable of Scheme 5.

In Figure 12, Scheme 6, the second train skips the third and fifth stations, the third train skips the second and fourth stations, and the fourth train skips the third station. The adjustment of train departure interval is also the same as Scheme 3. Scheme 5 resumes its planned operation from the eighth train.

Figure 12. Timetable of Scheme 6.

In Figure 13, Scheme 7 indicates all trains need to skip six stations. Specifically, the second train skips the third and fifth stations, the third train skips the second and fourth stations, the fourth train skips the third station, and the fifth train skips the second station.
Due to delays, the departure interval between the second and third trains is adjusted to 370 seconds. We shorten the interval between the third and fourth trains to 100 seconds, the fourth and fifth to 135 seconds, and the fifth and sixth to 115 seconds. Then, trains 7, 8, 9 and 10 are restored to their planned status.

In Table 7, we can observe that fewer trains are rescheduled as all trains skip more stations, so more trains are able to operate normally on the planned schedule. Compared with Table 6, we find that when the number of adjusted trains changes, the line’s total delay time and the delayed passenger count fluctuate greatly in value. The reason for this is that the more skip-stop instances for all trains, the shorter the trains’ running times, the less likely the delay is to spread, and the more flexibility there is to adjust the trains’ departure intervals. These factors have combined to reduce the total number of trains that need to be adjusted. As a result, the trains return to normal operation in a shorter time and the overall delay time on the line is reduced. However, the number of delayed passengers increases as they are not served at these skipped stations.

Table 7. Comparison of train adjustment.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Trains Adjusting the Trajectory</th>
<th>Trains on Planned Trajectory</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2</td>
<td>2, 3, 4, 5, 6, 7, 8</td>
<td>9, 10</td>
</tr>
<tr>
<td>3, 4, 5</td>
<td>2, 3, 4, 5, 6, 7</td>
<td>8, 9, 10</td>
</tr>
<tr>
<td>6, 7</td>
<td>2, 3, 4, 5, 6</td>
<td>7, 8, 9, 10</td>
</tr>
</tbody>
</table>

Figure 13. Timetable of Scheme 6.

6.3. Comparisons for WOA and ECM

In order to clearly validate the effectiveness of the proposed Pareto-based GA, we compared it with two algorithms, i.e., WOA and ECM. Key code of ECM can be found in Appendix.

6.3.1. Comparison between Pareto-Based GA and WOA

We used a whale optimization algorithm (WOA) to solve the same case above as a comparison. We depicted the convergence curve obtained by Pareto-based GA and WOA in Figure 14. The result indicates that the convergence speed of the former is faster than
that of the latter. When the total number of passengers is about 660 pax, the total delay time optimized by Pareto-based GA is 250 seconds less than that obtained by WOA—as much as 12.02%. When the total number of passengers is about 1200 pax, the total delay time optimized by Pareto-based GA is 175 seconds less than that obtained by WOA—13.10%.

**Figure 14.** Convergence curve of different algorithms.

We compare the objective function values obtained by the two algorithms. The overall costs when $P_1$ and $P_2$ are both 1 and the computing time are shown in Table 8. Through observation, the solution time of WOA is generally longer than that of the Pareto-based GA. In the seven schemes, the solution time of WOA was 1.96%, 0.87%, 1.11%, 3.82%, 4.72%, 7.2% and 10.24% longer than that of Pareto-based GA, respectively. It can be found that the more skips, the more the solving speed of WOA lags behind that of Pareto-based GA. In most schemes, the overall costs obtained by WOA are higher. For example, in Schemes 3, 4, 5 and 6, the overall costs obtained by WOA are 5.64%, 6.20%, 0.69% and 5.56% higher than that obtained by Pareto-based GA, respectively. When the normal operation of the train is restored only by adjusting the scheduled headway of the trains, the optimization effect of WOA is a bit better, as shown in the result of Scheme 1. However, when we use the skip-stop strategy, as shown in Scheme 4 and 6, the values of the two objective functions obtained by using WOA are larger and the overall costs are higher. In Scheme 4, the WOA optimization results in the total number of delayed passengers being 6.04% higher and the total delay time 6.36% longer than that obtained by Pareto-based GA, resulting in overall costs being 6.20% higher. In Scheme 6, the WOA optimization leads to the total number of delayed passengers being 4.47% higher and the total delay time 6.36% longer, resulting in overall costs being 6.20% higher compared to Pareto-based GA. However, in Scheme 5, WOA reduces the total delay time by 5.98% and the overall costs are still 6.20% higher. In Scheme 3, although the two algorithms obtain the same optimization results, WOA requires 1.11% longer solving time. Finally, Scheme 1 and 2, the WOA optimization results in a reduction in the overall costs by 34.91% and 0.80%, respectively. As summarized, Scheme 1 is only adjusted for departure intervals, while Scheme 2 is allowed to skip only one station. As a result, the cost reduction effect in Scheme 2 is not substantial. In overall skip-stop scenarios, the lowest overall costs of the optimized scheme using Pa-
reto-based GA are 2% lower than that of WOA. Thus, the Pareto-based GA is more suitable for using the skip-stop strategy to restore the normal operation of trains and achieving an optimization result.

**Table 8. Comparison of the results of different algorithms.**

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Number of the Skipped Stations of all Trains</th>
<th>Pareto-Based GA</th>
<th>WOA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of Delayed Passengers (Unit: Pax)</td>
<td>Total Delay Time of the Line (Unit: Second)</td>
<td>Overall Costs (Unit: $)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>2080</td>
<td>2713</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1832</td>
<td>2620</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1532</td>
<td>2596</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1407</td>
<td>2570</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1306</td>
<td>(minimum)</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>1203</td>
<td>2598</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>1154</td>
<td>2697</td>
</tr>
</tbody>
</table>

6.3.2. Comparison between Pareto-Based GA and ECM

We further compare Pareto-based GA with an epsilon constraint method (ECM). ECM transforms multi-objective problems into single-objective problems for optimization. The Pareto front is generated by iteratively updating the value of target boundary $\epsilon$.

There are two situations to state a priori. Situation 1 is where the first objective is selected as the main objective and the second objective is added as the constraint. Then, the bi-objective model proposed in Section 4.1 can be reformulated as Model (27).

$$
\begin{align*}
\text{Min} & \quad T \\
\text{s.t.} & \quad p \leq \varepsilon_1 \\
& \quad P_{\text{min}} \leq \varepsilon_1 \leq P_{\text{max}} \\
& \quad \text{Eq. (3) to (23)}
\end{align*}
$$

Referring to [25], we consider three values $a,b,c$ for $\varepsilon_1$ as follows: $a = \frac{P_{\text{min}} + P_{\text{max}}}{2}$, $b = \frac{a + P_{\text{min}}}{2}$, $c = \frac{a + P_{\text{max}}}{2}$. We compare the Pareto front generated by Pareto-based GA and ECM for solving the same case. The results are as shown in Figure 15.

![Figure 15. The Pareto front of different algorithms in Situation 1.](image-url)
In Figure 15, Pareto-based GA generates more Pareto solutions (50) than ECM (39), about a 28% improvement. The total delay time (2000 seconds) in the optimized solution obtained by ECM is longer than that by Pareto-based GA (1150 seconds). Here, delay time savings may amount to 42.5%. In particular, the maximum difference can reach about 1853 seconds as the number of delayed passengers is about 1178 pax.

In terms of Situation 2, the second objective is selected as the main objective, and the first objective is added as the constraint, and the bi-objective model can be reformulated as Model (27).

\[
\begin{align*}
\text{Min} & \quad P \\
\text{s.t.} & \quad T \leq \varepsilon_2 \\
& \quad T_{\text{min}} \leq \varepsilon_2 \leq T_{\text{max}} \\
Eq & \quad (3) \text{ to } (23)
\end{align*}
\]

We consider the three values \(d, e, f\) for \(\varepsilon_2\) as follows: \(d = \frac{r_{\text{min}} + r_{\text{max}}}{2}\), \(e = \frac{d + r_{\text{min}}}{2}\), \(f = \frac{d + r_{\text{max}}}{2}\). We compare the Pareto front generated by the two algorithms in Figure 16.

![Pareto front](image)

**Figure 16.** The Pareto front of different algorithms in Situation 2.

In Figure 16, Pareto-based GA generates more Pareto solutions (50) than ECM (32) with about a 56% improvement. The number of delayed passengers (1687 pax) in the optimization obtained by ECM is larger than that by Pareto-based GA (1607 pax). The number of delayed passengers is reduced by 4.74% approximately. When the total delay time is about 2430 seconds, the total number of delayed passengers obtained by ECM is more than that by Pareto-based GA, reaching 1230 pax, depicted as the red line in Figure 16.

As shown in Figures 15 and 16, we compared the computation times of Pareto-based GA and ECM for situations 1 and 2, respectively. In Situation 1, the running time for the Pareto-based GA was 215.21 seconds, while the ECM took 193.02 seconds. In Situation 2, the Pareto-based GA required 174.01 seconds, whereas the ECM needed 140.69 seconds. ECM is faster than Pareto-based GA with a 10.31% and 19.15% improvement in running time, respectively. In summary, Pareto-based GA and ECM yield better results for efficiency and effectiveness, respectively.
6.4. Sensitivity Analysis

A comprehensive sensitivity analysis can help us effectively understand the performance of the model under different conditions. We first examine the effect of adjusting the scheduled headway $H_2$, and then evaluate the effect of different combinations of penalty coefficients $P_1$ and $P_2$ on the dual objectives of minimizing the total train delay time. Last, the total number of delayed passengers is analyzed.

6.4.1. The Scheduled Headway $H_2$

As shown in Table 9, we set $H_2$ to seven different values and calculate the values of the two objective functions. The total number of skipped stations for all trains is four so that we can evaluate the model’s adaptability and effectiveness in response to changes in $H_2$.

<table>
<thead>
<tr>
<th>$H_2$ (Unit: Second)</th>
<th>Total Number of Delayed Passengers (Unit: Pax)</th>
<th>Total Delay Time of the Line (Unit: Second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>1642 (↑ 32.21%)</td>
<td>5937 (↑ 354.59%)</td>
</tr>
<tr>
<td>140</td>
<td>1731 (↑ 39.37%)</td>
<td>4201 (↑ 221.67%)</td>
</tr>
<tr>
<td>160</td>
<td>1862 (↑ 49.92%)</td>
<td>2465 (↑ 88.74%)</td>
</tr>
<tr>
<td>180</td>
<td>1242</td>
<td>1306</td>
</tr>
<tr>
<td>(basic)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>1599 (↑ 28.74%)</td>
<td>923 (↓ 29.33%)</td>
</tr>
<tr>
<td>220</td>
<td>1707 (↑ 37.44%)</td>
<td>746 (↓ 42.88%)</td>
</tr>
<tr>
<td>240</td>
<td>2066 (↑ 66.34%)</td>
<td>615 (↓ 52.91%)</td>
</tr>
</tbody>
</table>

If $H_2$ continues to increase, the interference between them decreases. The delay impact of the preceding train on the later trains is reduced, which in turn reduces the accumulated delay time. Thus, a larger headway allows ample buffer time for available trains to handle the delay, thereby reducing the total delay time.

In detail, as $H_2$ increases, the train’s departure frequency decreases and the overall transportation capacity declines. In this case, there is a challenge in dealing with peak passenger flows, with the number of delayed passengers increasing. However, as $H_2$ increases, the distance between trains becomes larger, the interference between vehicles decreases, and operational stability improves. When $H_2$ reaches 180 seconds, it achieves a balance. This headway not only ensures that the mutual impact between trains is reduced but also prevents passengers from waiting for excessively long periods. Consequently, at this interval, the number of delayed passengers is minimized. When $H_2$ increases again, the number of delayed passengers rises, which could be due to the longer waiting times for them caused by the extended headway.

As $H_2$ gradually increases from 120 seconds to 240 seconds, the minimum total number of delayed passengers and the minimum total delay time of the line show significant fluctuations with the change in $H_2$. Based on the comparisons, it is appropriate when $H_2$ =180.

6.4.2. Penalty Coefficient

We set the different values of penalty coefficient $P_1$ (unit: $/second) and $P_2$ (unit: $/pax) to perform the sensitivity analysis. Values of $P_1$ and $P_2$ are chosen from 1, 10, and 100, resulting in a total of nine different combinations corresponding to the nine distinct scenarios in Table 10. For each scenario, we calculate the Passenger Costs, Operator Cost, and Overall Costs for the seven proposed schemes. By comparing Overall Costs, we investigate the saving percentage. Table 10 displays the specific values.
Table 10. The performance of seven schemes in different scenarios.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Scheme</th>
<th>Passenger Costs</th>
<th>Operator Costs</th>
<th>Overall Costs</th>
<th>Overall Costs Saving Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>633</td>
<td>2080</td>
<td>2713</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>788</td>
<td>1832</td>
<td>2620</td>
<td>3.43%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1064</td>
<td>1532</td>
<td>2596</td>
<td>4.31%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1163</td>
<td>1407</td>
<td>2570</td>
<td>5.27%</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1242</td>
<td>1306</td>
<td>2548</td>
<td>6.08%</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1395</td>
<td>1203</td>
<td>2598</td>
<td>4.24%</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>1543</td>
<td>1154</td>
<td>2697</td>
<td>0.59%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Scheme</th>
<th>Passenger Costs</th>
<th>Operator Costs</th>
<th>Overall Costs</th>
<th>Overall Costs Saving Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>633</td>
<td>20800</td>
<td>21433</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>788</td>
<td>18320</td>
<td>19108</td>
<td>10.85%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1064</td>
<td>15320</td>
<td>16384</td>
<td>23.56%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1163</td>
<td>14070</td>
<td>15233</td>
<td>28.93%</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1242</td>
<td>13060</td>
<td>14302</td>
<td>33.27%</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1395</td>
<td>12030</td>
<td>13425</td>
<td>37.36%</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>1543</td>
<td>11540</td>
<td>13083</td>
<td>38.96%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Scheme</th>
<th>Passenger Costs</th>
<th>Operator Costs</th>
<th>Overall Costs</th>
<th>Overall Costs Saving Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>633</td>
<td>208000</td>
<td>208633</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>788</td>
<td>183200</td>
<td>183988</td>
<td>11.81%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1064</td>
<td>153200</td>
<td>154264</td>
<td>26.06%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1163</td>
<td>140700</td>
<td>141863</td>
<td>32.00%</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1242</td>
<td>130600</td>
<td>131842</td>
<td>36.81%</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1395</td>
<td>120300</td>
<td>121695</td>
<td>41.67%</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>1543</td>
<td>115400</td>
<td>116943</td>
<td>43.95%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Scheme</th>
<th>Passenger Costs</th>
<th>Operator Costs</th>
<th>Overall Costs</th>
<th>Overall Costs Saving Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1</td>
<td>6330</td>
<td>2080</td>
<td>8410</td>
<td>49.28%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>7880</td>
<td>1832</td>
<td>9712</td>
<td>41.44%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>10640</td>
<td>1532</td>
<td>12172</td>
<td>26.60%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>11630</td>
<td>1407</td>
<td>13037</td>
<td>21.39%</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>12420</td>
<td>1306</td>
<td>13726</td>
<td>17.23%</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>13950</td>
<td>1203</td>
<td>15153</td>
<td>8.63%</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>15430</td>
<td>1154</td>
<td>16584</td>
<td>—</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Scheme</th>
<th>Passenger Costs</th>
<th>Operator Costs</th>
<th>Overall Costs</th>
<th>Overall Costs Saving Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>63300</td>
<td>2080</td>
<td>27130</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>78800</td>
<td>1832</td>
<td>26200</td>
<td>3.43%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>106400</td>
<td>1532</td>
<td>25960</td>
<td>4.31%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>116300</td>
<td>1407</td>
<td>25700</td>
<td>5.27%</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>124200</td>
<td>1306</td>
<td>25480</td>
<td>6.08%</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>139500</td>
<td>1203</td>
<td>25980</td>
<td>4.24%</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>154300</td>
<td>1154</td>
<td>26970</td>
<td>0.59%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Scheme</th>
<th>Passenger Costs</th>
<th>Operator Costs</th>
<th>Overall Costs</th>
<th>Overall Costs Saving Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1</td>
<td>63300</td>
<td>2080</td>
<td>65380</td>
<td>57.92%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>78800</td>
<td>1832</td>
<td>80632</td>
<td>48.13%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>106400</td>
<td>1532</td>
<td>107932</td>
<td>30.57%</td>
</tr>
</tbody>
</table>

Table 10. The performance of seven schemes in different scenarios.
By observing the nine scenarios, we find that in Scenarios 1, 5, and 9, Scheme 5 has the lowest Overall Costs, but Scheme 1 achieves the highest. In Scenarios 2, 3, and 6, Scheme 7 has the lowest Overall Costs, while Scheme 1 has the highest. In Scenarios 4, 7, and 8, Scheme 1 has the lowest Overall Costs, while Scheme 7 has the highest. Due to the possibility of Schemes 1, 5, and 7 having the lowest Overall Costs, it is necessary to analyze different scenarios to determine their specific adoption conditions.

In Scenarios 1, 5, and 9, where \( P_1 \) and \( P_2 \) have the same values, analysis is conducted to see if the penalty levels for the count of delayed passengers and overall delay time traded off equally. This means that we treat operators’ interests and passenger satisfaction equally. Scheme 1 has the lowest Passenger Costs but the highest Operator Costs, while Scheme 7 has the lowest Operator Costs but the highest Passenger Costs. Both of them can only achieve minimization in one objective function, and their Overall Costs cannot be minimized; therefore, neither of them qualifies as the optimized scheme. On the other hand, Scheme 5, which balances the consideration of both costs, represents the optimized scheme with the lowest Overall Costs. Its Overall Costs is approximately 6.08% lower than the highest obtained from Scheme 1.

In Scenarios 2 and 6, where \( P_2 = 10P_1 \), we have imposed a more severe penalty on the line’s delay time, placing greater emphasis on reducing the number of delayed passengers. Since Scheme 1 has the longest total delay time in these scenarios, it has the highest Operator Costs after multiplying \( P_2 \) with total delay time and Overall Costs also becomes the highest. On the other hand, Scheme 7 has the fewest delayed passengers, leading to the lowest Passenger Costs and the lowest Overall Costs. Overall Costs of Scheme 7 is approximately 38.96% lower than Scheme 1. Scenario 3, where \( P_2 = 100P_1 \), further increases the penalty on the total delay time compared to Scenario 2. This widens the gap between the highest and lowest Overall Costs to $91,690. Ultimately, Scheme 7 achieves a reduction of approximately 43.95% in Overall Costs compared to Scheme 1, which highlights the significant advantage of Scheme 7.

In Scenarios 4 and 8, where \( P_2 = 0.1P_1 \), we prioritize the reduction in total delay time and impose a more severe penalty on the number of delayed passengers. Therefore, in these two scenarios, Scheme 7 has the most passengers who are delayed, which leads to the highest Passenger Costs after multiplying \( P_1 \) with the quantity of passengers who are delayed and subsequently the highest Overall Costs. On the other hand, Scheme 1 has the shortest total delay time, leading to the lowest Operator Costs and the lowest
Overall Costs. Scheme 1 reduced the total cost by approximately 57.92% compared to Scheme 7. In Scenario 7, where $P_\text{delay} = 0.01P_\text{no_delay}$, it further increases the penalty on the number of delayed passengers compared to Scenario 4. This widens the gap between the highest and lowest Overall Costs to $90,074. Ultimately, Scheme 1 achieves a reduction of approximately 57.92% in Overall Costs compared to Scheme 7, highlighting the significant advantage of Scheme 1.

Based on the above performance of these seven schemes applied in the nine scenarios, we can draw the conclusions that when balancing the interests of the operators and passenger satisfaction, if we consider them equally important, we can prioritize Scheme 5 for adjusting train operations. If we focus on reducing the total delay time on the line, Scheme 7 is the preferred choice. Conversely, if reducing the number of delayed passengers is deemed more important, Scheme 1 can be prioritized.

7. Conclusions

Initial unscheduled delays in urban rail transit not only considerably disrupt trains' punctuality but also lead to passengers' inconvenience. We propose a bi-objective mixed-integer linear programming model that uses the skip-stop operation to restore the scheduled plan. The model involves skip-stop decision variables and operation constraints derived from safety headway, loading capacity, and timetabling. The objective is to determine a well-designed skip-stop pattern to address the delays. In order to verify the validity of our proposed model, a Pareto-based genetic algorithm is employed in the case study to calculate the specific adjustment plans of all trains. With comparisons for WOA and ECM, the outcome mirrors the applicability of Pareto-based GA. Simultaneously considering operation recovery and passenger satisfaction, operators can select a suitable train operation adjustment scheme from the Pareto fronts.

Consequentially, with an increment in the collective count of skipped stations of all trains from 0 to 6, the number of trains requiring adjustments decreased from 7 to 5, the total delay time of the line reduced by 44.52% and the number of delayed passengers increased by 1.44 times. This is because skip-stopping saves dwelling times at intermediate stations, then reduces the overall running times, and finally reduces the total delay time of the line, but it also causes inconvenience to passengers who plan to get on and off at skipped stations, thus causing a rise in the number of delayed passengers. Furthermore, we conducted a sensitivity analysis by assigning penalty coefficients to the two objective functions. The results indicate that in seeking a trade-off between the operators’ interests and passenger satisfaction, we consider them equally important, so prioritize Scheme 5 for adjusting train operations with a maximum on overall cost saving percentage of as much as 6.08%. If we pay more attention to the operators’ interests, Scheme 7 is the preferred choice. Conversely, if we give passengers’ satisfaction a priority, Scheme 1 should be a superior choice.

Our study aimed to execute follow-up adjustment plans for handling issues of passenger congestion caused by URT delays. Skip-stop operations can restore trains’ plans and eliminate the delays, but will inevitably bring great inconvenience to passengers at skipped stations. The model is to balance their respective interests to recommend appropriate skip-stop schemes. Furthermore, sensitivity analysis help operators to analyze parameters’ impacts.

Our follow-up study will mainly cover the following areas.

1. Investigate the formulation of skip-stop schemes under dynamic passenger flows to improve the accuracy of the model.

2. Extend the study to larger and more intricate transit networks, which can help evaluate the scalability and application of the proposed model in various contexts.

3. Incorporate external factors, such as weather conditions, unexpected incidents, and infrastructural constraints, into the optimization model to enhance its robustness and effectiveness in real-world scenarios.
Author Contributions: Conceptualization, Y.W., Z.C. and S.Z.; methodology, Y.W. and Z.C.; software, Y.W.; validation, Y.W.; formal analysis, Y.W. and Z.C.; investigation, Y.W.; resources, Y.W.; data curation, Y.W.; writing—original draft preparation, Y.W.; writing—review and editing, Z.C.; visualization, Y.W.; supervision, Z.C. and S.Z.; project administration, Z.C., Z.Y., C.C. and S.Z.; funding acquisition, Z.C., Z.Y., C.C. and S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by National Natural Science Foundation of China (72101126) and (72101127), by the Science and Technology Project of Jiangsu Province, China (BK20200978), by Technology and Equipment of Rail Transit Operation and Maintenance (2022YW004), by CCF-Tencent Open Fund (CCF-Tencent IAGR20220111), by Guangdong Science and Technology Strategic Innovation Fund (the Guangdong–Hong Kong-Macau Joint Laboratory Program (2020B1212030009)), and by Undergraduate Innovation Training Program of Nantong University (202310304147Y) and (2023103041110).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix 1. Key Code of ECM.

```matlab
%% epsilon -Objective 1
para_aim_limit=1600;
%% Start optimizing
num_r=length(para_num_man);
num_R=num_r-2;
dim=num_R*para_num_car;
GA_size=150;
GA_gen=100;
GA_n=dim;
GA_long=20;
GA_gap=0.9;
FieldD=repmat([20
0.000001
0.999999
1
0
1
1],1,GA_n);
GA_x=crtbp(GA_size,GA_n*GA_long);
value=bs2rv(GA_x,FieldD);
y_1=fun_epsilon_1(value);
counter=0;
record_x=mean(y_1);
while counter<GA_gen
counter=counter+1;
shijing=ranking(y_1);
GA_select=select('sus',GA_x,shijing,GA_gap);
GA_select=recombin('xovsp',GA_select);
GA_select=mut(GA_select);
value=bs2rv(GA_select,FieldD);
y_2=fun_epsilon_1(value);
[GA_x y_1]=reins(GA_x,GA_select,1,1,y_1,y_2);
temp=bs2rv(GA_x,FieldD);
```
[min_value min_index]=min(y_1);
GA_save(counter,:)=\{temp(min_index,:) min_value\};
record_x(end+1)=mean(y_2);
counter
end
for i=2:length(record_x)
if record_x(i)>record_x(i-1)
record_x(i)=record_x(i-1);
end
end
temp=min(find(record_x<10^8));
record_x(1:temp)=record_x(temp);
final_x=GA_save(end,1:end-1);

y_1=round(y_1);
yy=y_2;
if sum(X_1(:))>limit_num
yy=yy+10^9*abs(sum(X_1(:))-limit_num);
end
if y_1>para_aim_limit
yy=yy+10^9*abs(y_1-para_aim_limit);
end
yyy(kkk,1)=yy;

%% epsilon -Objective 2
para_aim_limit=2550;
%% Start optimizing
num_r=length(para_num_man);
num_R=num_r-2;
dim=num_R*para_num_car;
GA_size=300;
GA_gen=100;
GA_n=dim;
GA_long=20;
GA_gap=0.9;
FieldD=repmat([20
0.000001
0.999999
1
0
1
1],1,GA_n);
GA_x=crtbp(GA_size,GA_n*GA_long);
value=bs2rv(GA_x,FieldD);
y_1=fun_epsilon_2(value);
counter=0;
while counter<GA_gen
counter=counter+1;
shiying=ranking(y_1);
GA_select=select('sus',GA_x,shiying,GA_gap);
GA_select=recombin('xovsp',GA_select);
GA_select=mut(GA_select);
value=bs2rv(GA_select,FieldD);
y_2=fun_epsilon_2(value);
[GA_x y_1]=reins(GA_x,GA_select,1,1,y_1,y_2);
temp=bs2rv(GA_x,FieldD);
[min_value min_index]=min(y_1);
GA_save(counter,:)=temp(min_index,:)
record_x(end+1)=mean(y_2);
counter
end
for i=2:length(record_x)
if record_x(i)>record_x(i-1)
record_x(i)=record_x(i-1);
end
end
final_x=GA_save(end,1:end-1);

y_1=round(y_1);
yy=y_1;
if sum(X_1(:))>limit_num
yy=yy+10^9*abs(sum(X_1(:))-limit_num);
end
if y_2>para_aim_limit
yy=yy+10^9*abs(sum(X_1(:))-limit_num);
end
yyy(kkk,1)=yy;

References


29. Yang, L.; Qi, J.; Li, S.; Gao, Y. Collaborative optimization for train scheduling and train stop planning on high-speed railways. Omega 2016, 64, 57–76.


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.