

Article

Optimization of Stochastic Robotaxi Dispatch Problem with Boarding-Time Recommendation

Yueqi Zhou ¹, Quanlu Xie ², Yangyang Peng ² and Liang Qi ^{2,*}¹ College of Transportation, Shandong University of Science and Technology, Qingdao 266590, China; chenxincomputer@sdust.edu.cn² College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China; xiequanlu3708@163.com (Q.X.); pengyangyanglm@163.com (Y.P.)

* Correspondence: qiliangsdkd@163.com

Abstract

This work investigates a stochastic robotaxi dispatch problem with boarding-time recommendation. Under an environment with stochastic travel time, a multi-objective stochastic programming model is formulated with the objectives of minimizing passenger experience penalty costs and minimizing total vehicle travel distance, allowing the platform to recommend appropriate boarding times to users. To solve this model efficiently, a clustering-based heuristic initialization method is designed according to the spatial distribution of requests to generate high-quality initial populations, iterated local search is employed to intensively explore promising solutions to improve algorithmic convergence, and a stochastic simulation method is introduced to evaluate the obtained solutions. On this basis, a multi-objective evolutionary algorithm with an iterated local search method (MOEA-ILSM) is proposed. Test instances are constructed based on real datasets from Didi Chuxing and the road network of Chengdu, and comparative experiments are conducted against three representative multi-objective optimization algorithms. The results demonstrate that the proposed algorithm outperforms the comparison algorithms in solution quality. Ablation experiments further validate the effectiveness of the boarding-time recommendation mechanism, showing that it can reduce passenger waiting time and improve the ride-sharing rate of requests.

Keywords: robotaxi dispatch; boarding-time recommendation; multi-objective optimization

1. Introduction

The great advancements in wireless communication, embedded systems, navigation technologies, sensors, and self-organizing networks have laid the foundation for the development of autonomous vehicles [1]. These vehicles, also known as self-driving cars, are computer-controlled cars capable of perceiving their environment, making decisions, and operating entirely without human intervention. Therefore, the development and application of autonomous vehicle technology can reduce traffic accidents and enhance vehicle safety [2]. In the near future, robotaxis are expected to become important components of urban mobility, offering enhanced safety, operational efficiency, and environmental sustainability, thereby contributing positively to the quality of transportation systems.

Received: 18 May 2026

Revised: 21 June 2026

Accepted: 24 June 2026

Published: 27 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Recently, the global deployment of advanced autonomous taxi services has increased significantly, although human drivers are still required to remain onboard [3]. Furthermore, several companies have begun deploying fully driverless robotaxi services on public roads, indicating the gradual transition of autonomous mobility services from pilot testing to real-world operations [4]. Robotaxis exhibit several key distinctions from conventional human-driven taxis (HDTs) [5–8]. First, on ride-hailing platforms such as Uber, Lyft, and DiDi, HDT drivers often make decisions based on individual interest, which may conflict with the optimization of overall efficiency. Second, without human drivers, robotaxis allow for higher operational control and better integration into intelligent transportation systems, promoting both efficiency and carbon reduction. Finally, compared to human drivers, autonomous systems enable consistent and smooth driving behavior, which significantly enhances ride comfort for passengers.

In the future, road traffic is expected to feature coexisting human-driven and autonomous vehicles [9]. While robotaxis can regulate their speed to control travel time, external factors such as traffic congestion and the stochastic nature of human driver behavior introduce significant uncertainty into their actual travel time. This study proposes and solves a novel robotaxi scheduling problem that incorporates stochastic travel time while recommending more reasonable pick-up times to passengers in accordance with their willingness. The proposed framework is particularly suitable for urban environments characterized by moderate-to-high travel demand, frequent ride-sharing opportunities, and uncertain traffic conditions, where limited adjustments to passenger pick-up times can significantly improve vehicle utilization and matching efficiency.

An effective dispatch model for HDTs must efficiently match vehicles with passengers to provide reliable, door-to-door service in urban areas [10,11]. In practice, HDT drivers depend on personal experience or ride-hailing platforms to search for passengers and may cancel ride requests assigned by the platforms, with the primary objective of maximizing their individual profit. Such uncoordinated, self-interested decision-making often decreases the overall matching quality. To address this issue, researchers have formulated various mathematical models to increase the number of served passengers while reducing taxi operational costs. For example, Miao et al. [12] introduce a data-driven robust dispatch framework and solved it via the minimax theorem, effectively minimizing discrepancies in the demand–supply ratio and reducing overall idle mileage. Kim et al. [13,14] reformulated the multi-objective taxi dispatch problem as a network flow model and applied a minimum-cost maximum-flow algorithm to maximize passenger service rates while simultaneously minimizing vehicle repositioning costs. Duan et al. [15] presented a taxi dispatch strategy applying a convex optimization, accommodating both immediate and reservation requests. However, the aforementioned approaches are typically suited for small-scale dispatch scenarios and often face limitations in computational efficiency. To tackle this issue, a variety of heuristic algorithms have been designed for taxi dispatch problems [16]. For instance, Zhu et al. [17] proposed a path-planning method based on a greedy algorithm, aiming to serve as many passengers as possible while minimizing both the number of taxis used and operational costs. Due to the limitations of greedy strategies in optimizing overall system performance, Gong et al. [18] adopted a differential evolution algorithm to optimize the match between taxis and passengers. In recent years, the application of deep reinforcement learning (DRL) to robotaxi dispatch has attracted growing research interest. For instance, Rivière et al. [19] designed a DRL-based approach aiming to decrease expected passenger waiting time. Liu et al. [20] incorporated contextual information into DRL frameworks to improve the efficiency of vehicle reallocation. Nevertheless, these learning-based models often require retraining when the operational environment changes, which can be a time-consuming process.

In taxi dispatch problems, shared on-demand mobility systems are often termed “ride-sharing” or “carpooling” systems, where passengers with different but relatively similar routes share a vehicle [21,22]. Ride-sharing has enormous potential to benefit all participants and society. Alisoltani et al. [23] delved into the ride-sharing effect on road congestion and showed that the ride-sharing system can help significantly alleviate traffic congestion based on a large-scale simulation in Lyon. Ke et al. [24] pointed out that the ride-sharing market has a larger service rate than the traditional no-pooling market and showed that ride-sharing can reduce the time cost of both ride-sharing passengers and drivers in the transport system under some scenarios. Huang et al. [25] developed an ant-path-based allocation algorithm aimed at maximizing both the number of served passengers and vehicle seat utilization. Li et al. [26] presented a data-driven robust optimization method that combines gated recurrent unit-based travel-time predictions with a single-stage robust optimization framework to tackle carpooling challenges with stochastic travel time. However, carpooling may increase passengers’ travel time, as vehicles often need to take longer detour routes to serve all passengers. Additionally, passengers already on board may experience further delays while the vehicle waits at pick-up points for other passengers to board. Additionally, due to the uncoordinated boarding time (BT) specified by passengers, those already on board may experience further delays as the vehicle waits at pick-up points for others to board. The BT of passengers plays a critical role in robotaxi dispatch systems and is commonly considered a key constraint in previous research. Huang et al. [27] incorporated a time window to constrain passengers’ BT in the carpool service problem and applied an evolutionary algorithm to solve the problem. Guo et al. [28] proposed a ride-sharing framework featuring dynamic timeframe segmentation and anticipation-based commuter migration to improve the quality of taxi–passenger matching. Recognizing the impact of traffic conditions, Liang et al. [29] formulated a dial-a-ride problem with dynamic travel time to maximize system profit by optimizing the routes of robotaxis. Zhang et al. [30] proposed a co-prediction approach, aiming to assist drivers in locating passengers efficiently, thereby reducing the waiting time of passengers.

Recommending an appropriate BT within a reasonable timeframe can significantly improve the matching quality between taxis and passengers, thereby enhancing service efficiency and passenger satisfaction. Li et al. [31] further incorporated a boarding-time recommendation mechanism into taxi dispatching. Their findings demonstrated that recommending appropriate boarding times to passengers within a reasonable range can reduce passengers’ travel costs while improving the per-kilometer profit of taxis. However, the aforementioned study on boarding-time recommendation was based on the assumption of deterministic travel time, failing to adequately account for the uncertainty in travel time induced by factors such as traffic congestion. Although extensive research has been conducted on ride-sharing dispatching [17,22,28], travel-time uncertainty, congestion-aware operations [26,29], and passenger scheduling flexibility through pick-up/drop-off and boarding-time recommendations [22,31], these aspects have generally been investigated separately rather than within an integrated optimization framework. Consequently, the interaction between travel-time uncertainty and boarding-time recommendation remains insufficiently explored. To address this gap, this paper proposes a multi-objective robotaxi dispatching framework that jointly considers stochastic travel times and boarding-time recommendations, aiming to improve both system efficiency and passenger service quality. In practical robotaxi systems, the proposed mechanism can be implemented as an online decision-support module integrated into the dispatching platform. In addition, passenger non-compliance with recommended boarding times may introduce further uncertainty, which is considered implicitly in the stochastic formulation. The main contributions are listed as follows:

- (1) A mixed-integer programming model is established to formulate the concerned problem mathematically.
- (2) In order to tackle this problem, a heuristic method is designed to produce high-quality solutions. Then, we propose a multi-objective evolutionary algorithm with an iterated local search method (MOEA-ILSM). Furthermore, a stochastic simulation method is developed to evaluate the obtained solutions.

In addition, extensive experiments are conducted to assess the effectiveness of MOEA-ILSM through comparisons with three representative multi-objective optimization algorithms, namely, the non-dominated sorting genetic algorithm II (NSGA-II) [32], a multi-objective evolutionary algorithm based on decomposition (MOEA/D) [33], and the multi-objective artificial bee colony algorithm (MOABC) [34]. Moreover, a sensitivity analysis indicates that the developed model can effectively reduce passenger experience penalty costs and vehicle travel distance while enhancing passenger experience.

The remainder of this paper is structured as follows. Section 2 describes the investigated problem, develops its mathematical model, and states the proposed algorithm. Section 3 reports comparative experiments and analyzes the results. Finally, Section 4 summarizes the paper and outlines directions for future research.

2. Materials and Methods

2.1. Problem Formulation

This section presents a stochastic robotaxi dispatch problem with boarding-time recommendation (SRD-BTR), which is formulated as a multi-objective mixed-integer programming model. Let $N = \{0, 1, 2, \dots\}$ represents a set of all natural numbers, $N^+ = N \setminus \{0\}$, $N_m = \{0, 1, 2, \dots, m\}$, and $N_m^+ = N_m \setminus \{0\} = \{1, 2, \dots, m\}$. The notations used in this paper are defined in Table 1.

Table 1. Notations and definitions.

Indices and Parameters	
m	Total number of requests.
n	Total number of vehicles.
R	Set of all requests, where $R = \{r_1, \dots, r_m\}$, r_i denotes a request i , $i = 1, \dots, m$.
V	Set of all vehicles, where $V = \{v_1, \dots, v_n\}$, v_k denotes a vehicle k , $k = 1, \dots, n$.
u_i	Pick-up point of r_i .
o_i	Drop-off point of r_i .
τ_i	Initial boarding time (BT) of r_i .
q_i	Number of passengers in r_i .
s_k	Starting point of v_k .
e_k	Destination points of v_k .
P	Set of pick-up points and drop-off points of requests, starting points and destination points of vehicles, where
	$P = \left(\bigcup_{i=1}^m u_i \right) \cup \left(\bigcup_{i=1}^m o_i \right) \cup \left(\bigcup_{k=1}^n s_k \right) \cup \left(\bigcup_{k=1}^n e_k \right).$
$l_{\alpha,\beta}$	Shortest travel distance from point α to point β .
$t_{\alpha,\beta}$	Estimated travel time from point α to point β .
ξ_i	Requester's willingness for the recommended BT: $\xi_i = 1$ if r_i accept the recommendation; and otherwise, $\xi_i = 0$.
c_k	Capacity of v_k .
$\hat{\psi}$	Maximum absolute value of BT offsets measured in minutes.
λ_1	Penalty coefficient of waiting time of passenger.
λ_2	Penalty coefficient of BT offset.

λ_3	Penalty coefficient of detour rate.
F	A large positive constant.
Decision variables	
$x_{\alpha,\beta}^k$	A decision variable indicating whether v_k passes through point β immediately after point α .
ψ_i	A BT offset, the BT is $ \psi_i $ minutes earlier if $\psi_i < 0$; and otherwise, the BT is $ \psi_i $ minutes later.
a_α^k	Arrival time of v_k at point α .
b_α^k	Leaving time of v_k at point α .
d_α^k	Cumulative distance traveled by v_k when it arrives at point α .
h_α^k	Occupied seats in v_k when leaving point α .
w_i	Waiting time of passengers of r_i .
δ_i	Detour rate of r_i .

2.1.1. Problem Statement

A request represents a set of passengers to be transported from a pick-up point to a drop-off point. Let r_i denote a request, where u_i , o_i , and τ_i represent its pick-up point, drop-off point, and initial boarding time, respectively, and q_i represents the number of passengers of r_i . When r_i agrees with adjusting its initial boarding time, $\xi_i = 1$; otherwise $\xi_i = 0$. Additionally, ψ_i denotes a BT offset of r_i . The maximum allowable boarding-time offset is denoted by $\hat{\psi}$, which specifies the largest deviation between a passenger's initial boarding time and the recommended boarding time. Thus, passengers who accept boarding-time recommendations may be assigned an adjusted boarding time within the interval $[\tau_i - \hat{\psi}, \tau_i + \hat{\psi}]$. Let v_k denote a vehicle, where s_k represents its starting point, and c_k represents its capacity, i.e., the maximum number of passengers it can carry. The sets of requests and vehicles are denoted by $R = \{r_1, \dots, r_m\}$ and $V = \{v_1, \dots, v_n\}$, respectively, where m and n represent the total number of requests and vehicles, respectively. In the studied problem, the travel time of a vehicle between any two locations is not deterministic but is modeled as a random variable to capture the uncertainty inherent in real-world traffic conditions. Specifically, travel time is modeled as a left-truncated normal distribution, where the mean is determined by travel distance and average vehicle speed, and the standard deviation captures traffic-related uncertainty. The left truncation ensures non-negative travel-time realizations. This distribution provides a computationally tractable representation of travel-time uncertainty, enabling the evaluation of the proposed boarding-time recommendation mechanism under stochastic traffic conditions.

At the initial time of the scenario illustrated in Figure 1, all vehicles are idle and located at their respective initial positions, and no vehicle is currently assigned to an ongoing trip (all s_k and e_k are inactive). Therefore, all vehicles are available for immediate dispatch to incoming requests. Figure 1 intuitively illustrates the motivation of this study through a representative example. The figure depicts the positions of vehicles at 9:00 a.m., where the black, red, and blue lines represent the travel routes of vehicles v_1 , v_2 , and v_3 , respectively. Figure 1a shows the vehicle service routes without the introduction of boarding-time recommendations. The boarding time specified by requests r_1 , r_2 , and r_3 are 9:08 a.m., 9:06 a.m., and 9:01 a.m., respectively, and under this scenario, the three requests are served independently by vehicles v_1 , v_2 , and v_3 . Since vehicle v_3 is approximately 2 km away from the pick-up location of request r_3 , it is unable to arrive within 1 min, resulting in a considerably long waiting time for the passenger. Figure 1b shows the vehicle service routes after the introduction of boarding-time recommendations. Assuming that requests r_1 and r_3 agree to adjust their boarding time within the allowable adjustment interval (with a maximum offset of $\hat{\psi} = 15$ min), the boarding time of request r_1 is advanced by 6 min and that of request r_3 is delayed by 4 min. As a result, requests r_1 and r_2 can be served

together in a carpooling arrangement on vehicle v_1 , thereby reducing the number of vehicles in use by one and freeing up a vehicle to serve other requests. In this arrangement, user r_1 experiences two types of disutility, including an earlier boarding time (6 min in advance) and a longer in-vehicle time due to the detour for user r_2 , which are captured in our model as the time shift penalty ($\lambda_2 \cdot \xi_i \cdot \psi_i^i$) and the detour penalty ($\lambda_3 \cdot (\delta_i - 1)$), respectively. Although user r_1 bears these costs, this trade-off is necessary to achieve improved system-level efficiency under the multi-objective optimization framework. Meanwhile, delaying the boarding time of request r_3 by 4 min improves the matching between the recommended pick-up time and the estimated arrival time of vehicle v_3 . Consequently, the temporal mismatch is reduced, leading to a lower expected waiting time compared to Figure 1a. This example demonstrates that appropriately recommending passenger boarding times can not only enhance the feasibility of carpooling but also effectively improve the overall travel experience of passengers.

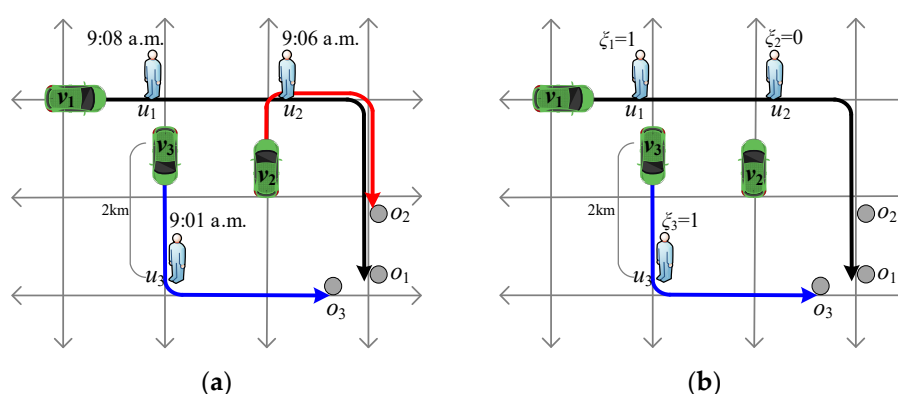


Figure 1. Illustration of the research motivation: (a) without boarding-time recommendation; (b) integrating boarding-time recommendation. The arrows in different colors indicate the routes assigned to different vehicles.

2.1.2. Mathematical Model

For request r_i , the adjusted boarding time is given by $\tau_i + \xi_i \cdot \psi_i$. When the actual arrival time of the vehicle at the pick-up point u_i is later than the passenger's adjusted boarding time, the passenger is required to wait at the pick-up point, and the waiting time is defined as follows:

$$w_i = \sum_{k \in N_a^+} \max(a_{u_i}^k - (\tau_i + \xi_i \cdot \psi_i), 0) \quad (1)$$

Since the travel time of vehicle v_k is a random variable, the arrival time of vehicle v_k at the pick-up point of request r_i , denoted by $a_{u_i}^k$, is also a random variable, which consequently renders the waiting time w_i a random variable as well.

Furthermore, adjustments to the boarding time of requests will have a certain impact on the passenger travel experience. The boarding time offset of request r_i can be quantified as $\xi_i \cdot |\psi_i|$. However, since the absolute value function is nonlinear, directly incorporating it into the mixed-integer programming model would destroy the linear structure of the model. To overcome this issue, a non-negative continuous auxiliary variable ψ_i^+ is introduced to replace $|\psi_i|$, which is characterized by the following linear constraints:

$$\psi_i^+ \geq \psi_i, \psi_i^+ \geq -\psi_i, \forall i \in N_m^+ \quad (2)$$

Driven by minimization of the objective function, the optimization process will force ψ_i^+ to take its minimum feasible value. When $\xi_i = 1$, ψ_i^+ will tighten to $|\psi_i|$; when $\xi_i = 0$, both $\xi_i \cdot |\psi_i|$ and $\xi_i \cdot \psi_i^+$ are equal to zero, and the equivalence trivially holds. Consequently, the original nonlinear term $\xi_i \cdot |\psi_i|$ can be equivalently linearized as $\xi_i \cdot \psi_i^+$. Considering that

passengers can generally only accept boarding-time adjustments within a limited range, the offset ψ_i is restricted to the interval $[-\hat{\psi}, \hat{\psi}]$, where $\hat{\psi}$ denotes the maximum allowable offset.

When the passenger of request r_i shares a vehicle with passengers of other requests, the vehicle may incur detours, leading to an increase in the actual travel distance for the passenger and consequently affecting passenger satisfaction. To quantify the degree of detour induced by carpooling, a detour ratio δ_i is introduced, defined as the ratio of the actual travel distance of the passenger of request r_i to the shortest travel distance between their pick-up and drop-off points [35], as given in the following equation:

$$\delta_i = \sum_{k \in N_n^+} \frac{d_{o_i}^k - d_{u_i}^k}{l_{u_i, o_i}} \quad (3)$$

where $d_{u_i}^k$ and $d_{o_i}^k$ denote the cumulative travel distances of vehicle v_k along its route upon arriving at the pick-up point u_i and drop-off point o_i of request r_i , respectively. Therefore, $(d_{o_i}^k - d_{u_i}^k)$ represents the actual travel distance experienced by request r_i between pick-up and drop-off along the realized shared route, including any intermediate pick-ups and drop-offs of other requests. l_{u_i, o_i} denotes the shortest travel distance from u_i to o_i .

A multi-objective optimization model for the studied problem is presented below, aiming to simultaneously optimize passenger experience and total vehicle travel distance.

$$\min f_1 = \lambda_1 \cdot \sum_{i \in N_n^+} E(w_i) + \lambda_2 \cdot \sum_{i \in N_m^+} \xi_i \cdot \psi_i^+ + \lambda_3 \cdot \sum_{i \in N_m^+} (\delta_i - 1) \quad (4)$$

$$\min f_2 = \sum_{k \in N_n^+} d_{e_k}^k \quad (5)$$

s.t. (1)–(3), and the following constraints:

$$\sum_{k \in N_n^+} \sum_{\beta \in P} x_{u_i, \beta}^k = \sum_{k \in N_n^+} \sum_{\alpha \in P} x_{\alpha, o_i}^k = 1, \forall i \in N_m^+ \quad (6)$$

$$\sum_{\beta \in P} x_{\alpha, \beta}^k = \sum_{\beta \in P} x_{\beta, \alpha}^k, \forall k \in N_n^+, \forall \alpha \in (\cup_{i \in N_m^+} u_i) \cup (\cup_{i \in N_m^+} o_i) \quad (7)$$

$$\sum_{\beta \in P} x_{u_i, \beta}^k = \sum_{\alpha \in P} x_{\alpha, o_i}^k, \forall k \in N_n^+, \forall i \in N_m^+ \quad (8)$$

$$\sum_{\beta \in P} x_{s_k^-, \beta}^k = 1 - \sum_{k \in N_n^+} \sum_{\alpha \in P} x_{\alpha, s_k^-}^k = 1, \forall \tilde{k} \in N_n^+ \quad (9)$$

$$\sum_{\alpha \in P} x_{\alpha, e_k}^k = 1 - \sum_{k \in N_n^+} \sum_{\beta \in P} x_{e_k, \beta}^k = 1, \forall \tilde{k} \in N_n^+ \quad (10)$$

$$\sum_{k \in N_n^+} \sum_{\beta \in P} x_{s_k^-, \beta}^k = \sum_{k \in N_n^+} \sum_{\alpha \in P} x_{\alpha, e_k}^k = 1, \forall \tilde{k} \in N_n^+ \quad (11)$$

$$\sum_{k \in N_n^+} x_{u_i, u_i}^k = \sum_{k \in N_n^+} x_{o_i, o_i}^k = 0, \forall i \in N_m^+ \quad (12)$$

$$E(a_\beta^k - b_\alpha^k - t_{\alpha, \beta}) + (1 - x_{\alpha, \beta}^k)F \geq 0, \forall k \in N_n^+, \forall \alpha, \beta \in P \quad (13)$$

$$b_{u_i}^k \geq \max\{a_{u_i}^k, \tau_i + \xi_i \cdot \psi_i^+\}, \forall k \in N_n^+, \forall i \in N_m^+ \quad (14)$$

$$b_{o_i}^k \geq a_{o_i}^k, \forall k \in N_n^+, \forall i \in N_m^+ \quad (15)$$

$$h_{u_i}^k - h_\alpha^k \geq q_i - (1 - x_{\alpha, u_i}^k) \cdot F, \forall k \in N_n^+, \forall i \in N_m^+, \forall \alpha \in P \quad (16)$$

$$h_\alpha^k - h_{o_i}^k \leq q_i + (1 - x_{\alpha, o_i}^k) \cdot F, \forall k \in N_n^+, \forall i \in N_m^+, \forall \alpha \in P \quad (17)$$

$$h_\alpha^k \leq c_k, \forall k \in N_n^+, \forall \alpha \in P \quad (18)$$

$$d_\beta^k - d_\alpha^k \geq l_{\alpha, \beta} - (1 - x_{\alpha, \beta}^k)F, \forall k \in N_n^+, \forall \alpha, \beta \in P \quad (19)$$

$$d_{o_i}^k \geq d_{u_i}^k, \forall k \in N_n^+, \forall i \in N_m^+ \quad (20)$$

$$x_{\alpha,\beta}^k \in \{0,1\}, \forall k \in N_n^+, \forall i \in N_m^+, \forall \alpha, \beta \in P \quad (21)$$

$$-\hat{\psi} \leq \psi_i \leq \hat{\psi}, \forall i \in N_m^+, d_{\alpha}^k \geq 0, d_{\alpha}^k \geq 0, h_{\alpha}^k \geq 0 \forall k \in N_n^+, \forall i \in N_m^+, \forall \alpha \in P \quad (22)$$

Objective function (4) minimizes the total penalty cost, which comprises the passenger waiting penalty cost, the boarding-time adjustment penalty cost, and the vehicle detour penalty cost, where $E(\cdot)$ denotes the expectation of a random variable. Objective function (5) minimizes the total travel distance of vehicles. Constraint (6) guarantees that each request can be served by one vehicle. Constraint (7) guarantees that a vehicle visiting a pick-up or drop-off point must subsequently leave it, i.e., flow conservation. Constraint (8) ensures that the pick-up and drop-off points of a request are visited by the same vehicle. Constraints (9), (10), and (11) denote that v_k departs from a starting point s_k and stops at a virtual destination point e_k , which is the same as the drop-off point of the last passenger to be delivered. Constraint (12) stipulates that the pick-up or drop-off point of a request should be visited only once. Constraint (13) formulates the time when vehicles arrive at each point. Constraints (14) and (15) define the time when vehicles leave each point. Constraints (16) and (17) provide the number of passengers in the vehicle after it departs from a point. Constraint (18) guarantees that the number of passengers in the vehicle does not exceed its capacity. Constraint (19) characterizes the cumulative travel distance of vehicles when arriving at each point. Constraint (20) ensures that the pick-up point of each request is visited prior to its drop-off point. Constraints (21) and (22) limit the range of decision variables.

2.2. Proposed Approach

Over the past few years, evolutionary algorithms have been progressively applied and enhanced to address a variety of NP-hard problems, with their effectiveness firmly established in the literature. Moreover, simulated annealing can be utilized to further optimize promising solutions. Given the multiple objectives and stochasticity of the problem under consideration, this study develops a multi-objective evolutionary algorithm with an iterated local search method (MOEA-ILSM) and a stochastic simulation method. Particularly, MOEA-ILSM employs a cooperative search method between population (named as $\mathcal{P}$) and external archive (named as $\mathcal{A}$), and the search information found by them is shared between them.

2.2.1. Solution Representation

We adopt a double-dimensional sequence $\pi = (\sigma; \chi)$ to represent the solution (called an individual as well). σ is a vehicle route sequence, indicating the vehicles' visiting order. The integer 0 divides σ into n subsequences, denoted by $\sigma_1, \dots, \sigma_n$, each of which represents a vehicle route. In subsequences, the pick-up points of requests are indexed as $1, \dots, m$, and the drop-off points of requests are indexed as $m+1, \dots, 2m$. It is obvious that the pick-up point and drop-off point of r_i are indexed as i and $m+i$, respectively. $\chi = (\chi_1, \dots, \chi_i, \dots, \chi_m)$ is a sequence of scaled BT offsets, where χ_i is a real number in the range $[-1, 1]$. The actual BT offset of r_i is computed as $\xi_i \cdot \chi_i \cdot \hat{\psi}$. Figure 2 depicts the solution for the instance in Figure 1, with $\hat{\psi}$ set to 10.

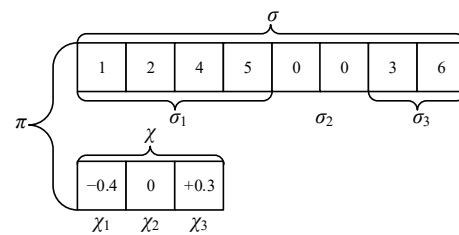


Figure 2. Illustration of solution representation.

2.2.2. Population Initialization

It is widely acknowledged that an initialized population with high-quality solutions enhances the effectiveness of algorithms. This study proposes a heuristic method to obtain high-quality solutions. The details are as follows.

Inspired by ref. [36], requests are clustered based on their pick-up and drop-off locations, and passengers of the requests within the same cluster are arranged for carpooling. The pseudo-code of the request clustering algorithm is provided in Algorithm 1. For each request r_i , r_i is added to the cluster centered on r_i if it meets the following four conditions (lines 8–11), where ζ_i denotes the vector from u_i to o_i : (1) the number of requests in this cluster is less than a threshold, denoted by θ ; (2) the distance from u_i and o_i to the line defined by the direction of vector $\zeta_{i'}$ is less than a given threshold, denoted by $\iota \cdot d_{\text{euc}}(u_{i'}, o_{i'})$, where ι is a coefficient, and $d_{\text{euc}}(u_{i'}, o_{i'})$ represents the Euclidean distance between $u_{i'}$ and $o_{i'}$; (3) the angle between vectors ζ_i and $\zeta_{i'}$ is less than a threshold, denoted by θ ; and (4) the projection of vector ζ_i onto the direction of vector $\zeta_{i'}$ intersects with vector $\zeta_{i'}$ (lines 14–17). To illustrate the request clustering more clearly, an example with 5 requests is displayed in Figure 3. The two black dashed lines are parallel to $\zeta_{i'}$ and are located at a distance of the distance threshold, i.e., $\iota \cdot d_{\text{euc}}(u_{i'}, o_{i'})$, from it. The center of a cluster is $r_{i'}$. Only r_1 satisfies all of the aforementioned conditions and is therefore eligible for inclusion in the cluster; the remaining requests do not. r_2 , r_4 , and r_5 violate the second, third, and fourth conditions, respectively.

Algorithm 1. Request Clustering Algorithm.

Input: Request set $R = \{r_1, \dots, r_m\}$.

Output: Request cluster set $R_c = \{R_1, \dots, R_\omega\}$.

```

1:  $R_c = \emptyset$ .
2:  $\omega = 0$ .
3:  $R' = R$ . // make a copy of  $R$ 
4: while ( $R'$  is not empty) do
5:   Select a request  $r_i$  from  $R'$  at random and remove it.
6:    $\text{add} = \text{false}$ .
7:   for  $j = 1$  to  $\omega$  do
8:     if ( $r_i$  meets the conditions to be added to the cluster centered at  $C_j$ ) then
9:        $R_j = R_j \cup \{r_i\}$ .
10:       $\text{add} = \text{true}$ .
11:      break.
12:   end if
13: end for
14: if  $\text{add} == \text{false}$  then
15:    $\omega = \omega + 1$ .
16:    $R_\omega = \{r_i\}$ 
17:    $C_\omega = r_i$ .
18: end if
19: end while

```

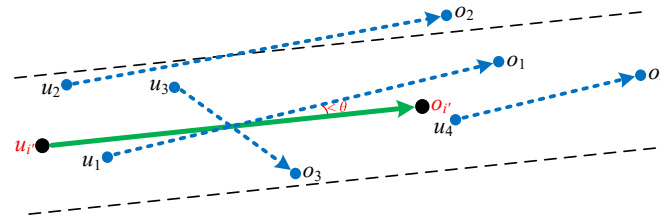


Figure 3. Example of request clustering.

After clustering requests, a visiting route is generated for serving requests within each cluster. For each non-center request in a cluster, the pick-up and drop-off points are orthogonally projected onto the vector defined by the cluster center's pick-up and drop-off points. The projected positions of the pick-up and drop-off points along the direction of the cluster center are then used to determine the visiting route. As illustrated in Figure 3, r_i and r_1 are grouped into the same cluster, with r_i acting as the center. u_1 and o_1 are projected onto the vector c_r . Based on the projection, the visiting route is u_i, u_1, o_1 . Subsequently, the clusters are sorted based on the earliest initial boarding time among requests within each cluster. Following this order, vehicles are assigned to the visiting routes of the clusters. A heuristic rule [37] is applied to sequentially assign the visiting routes of the sorted clusters to available vehicles. Assume that the first point in the visiting route of a cluster is u_i . Let $\bar{l}(*, u_i)$ denote the average distance from starting points of all available vehicles to u_i . The probability that the visiting route of the cluster is assigned to vehicle v_k is computed by:

$$pr(i, k) = \frac{\max\{\bar{l}(*, u_i) - l(s_k, u_i), 0\}}{\sum_{i \in N_m} \max\{\bar{l}(*, u_i) - l(s_k, u_i), 0\}} \quad (23)$$

In accordance with Equation (23), a route is more likely to be assigned to the nearest available vehicle. Nevertheless, there remains a possibility for the route to be assigned to an available vehicle that is farther away. It is worth noting that once a vehicle is assigned a route, it becomes unavailable for subsequent route assignments. If no vehicle is available, the visiting routes of the clusters are sequentially appended to the end of existing vehicle routes, such that the distance from the last point of a vehicle's current route to the first point of the cluster's route is minimized.

Through the above processes, the routes for vehicles are generated. Then, the BT offsets of requests are heuristically determined. For vehicle routes that serve only a single order, the scaled BT offset of the served order is randomly assigned a value between 0 and 1. For each vehicle route that serves multiple orders, a BT offset is assigned to each order sequentially according to the service order along the route. The BT offsets should be randomly generated while ensuring that the boarding time of each order is later than that of the previous one. If this condition cannot be satisfied, the scaled BT offset should instead be randomly assigned a value between 0 and 1.

To enhance the diversity of the initial population, half of the individuals are initialized using the aforementioned heuristic method, while the remaining half are generated through random initialization. The initialized individuals may violate the vehicle capacity constraints. To address this, for any vehicle route that exceeds the capacity limit, the pick-up points of the overloaded passengers are postponed accordingly.

2.2.3. Genetic Operations

In MOEA-ILSM, the population $\mathcal{P}$ employs genetic operations, including crossover and mutation operators, to search through the solution space. Each individual consists of a vehicle route sequence and a sequence of scaled BT offsets. Therefore, two crossover and two mutation operators are designed.

Crossover operators aim to generate new individuals using parent individuals. As demonstrated above, $\mathcal{A}$ preserves the best-performing individuals identified during the preceding search phase. To enable the exchange of search information between $\mathcal{P}$ and $\mathcal{A}$, for an individual π' in $\mathcal{P}$, another individual π'' is randomly selected from either $\mathcal{P}$ or $\mathcal{A}$. Let π' and π'' be parents. Figure 4 illustrates a procedure for generating a vehicle route sequence σ''' in a new individual. It is outlined as follows: Step (1): Two vehicle routes are randomly selected from each of the two individuals: σ_3 in σ' and σ_1 in σ'' are selected. Step (2): The selected vehicle routes in the two parent individuals are exchanged: σ_3 in σ' is exchanged with σ_1 in σ'' , and σ''' is obtained to replace σ_3 in σ' with σ_1 in σ'' . Step (3): The duplicated points from the other vehicle routes are deleted: point indices 1 and 4 are removed from σ_1 . Step (4): The unrouted points are checked and randomly inserted into other routes or formed into a new route with equal probability if there are idle vehicles. A new route is reconstructed and is assigned to v_2 . Then, in the new individual, the sequence of scaled BT offsets, denoted by χ''' , is generated based on χ' as follows: First, in χ' , the scaled BT offsets of requests in the selected route of σ' involved in Step (1) are replaced with the corresponding values in χ'' . Second, for each request involved in Step (4), their scaled BT offsets are assigned according to a BT offset assignment rule. In the vehicle route, when the passengers of a request boards the vehicle, if all other passengers assigned to the same vehicle have already been dropped off at their destinations, the scaled BT offset of that request is assigned a random value in $[0, 1]$; otherwise, it is assigned a random value in $[-1, 1]$.

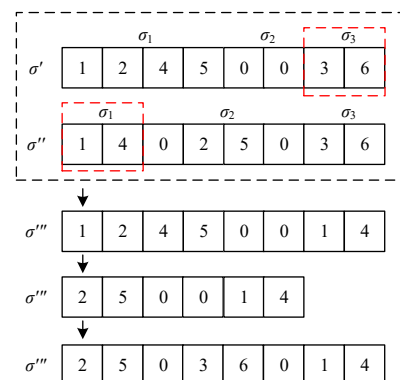


Figure 4. Illustration of the designed crossover operator for the vehicle route sequence.

The goal of mutation operators is to increase the population's diversity. After applying the crossover operation, a mutation operation is performed on the new individuals with a probability of p_m . For the vehicle route sequence, this study proposes five mutation operators, each selected with equal probability: (1) Select two requests from two different routes randomly and swap their pick-up and drop-off points. (2) Select a request at random on a route and insert its pick-up and drop-off points into random positions within another route. (3) Select two requests on a route randomly and swap their pick-up and drop-off points, respectively. (4) Select a request on a route randomly and reinsert its pick-up and drop-off points at new random positions within the same route. (5) Select two vehicles randomly and exchange their routes. For the sequence of scaled BT offsets, a vehicle is selected randomly, and the scaled BT offsets of the requests it serves are updated according to the BT offset assignment rule.

After applying genetic operations, $2N$ individuals are obtained by combining individuals in the current population and all of the new individuals. Then, the rank and crowding distance strategy is adopted to choose N individuals.

2.2.4. Iterated Local Search Method

In the proposed algorithm, the external archive $\mathcal{A}$ is maintained to store the non-dominated solutions obtained during the search process. To further enhance the quality of these solutions, inspired by the concept of simulated annealing [38], an iterated local search method (ILSM) is designed to explore the neighborhood of the non-dominated solutions stored in $\mathcal{A}$. The pseudocode of ILSM is presented in Algorithm 2, where $\text{rand}(0, 1)$ denotes a random number generated uniformly over the interval $[0, 1]$. For each solution in $\mathcal{A}$, a mutation operation is applied with probability $\gamma \cdot F_c / F_M$ to generate a neighborhood solution, where F_c and F_M denote the current and maximum numbers of fitness evaluations, respectively, and $\gamma \cdot$ is a given parameter. If the neighborhood solution dominates the current solution, the current solution is replaced by the neighborhood solution, and the search continues from the neighborhood solution as the new starting point, proceeding iteratively in this manner. For each solution selected with the aforementioned probability, at most $\varepsilon \cdot m$ mutation operations are performed for enhancement, where ε is a given parameter. Finally, the external archive $\mathcal{A}$ is updated using the set of neighborhood solutions ω generated throughout the search process.

Algorithm 2. ILSM.

Input: γ , ε , F_M , and external archive $\mathcal{A}$.

Output: $\mathcal{A}$.

```

1:   $\omega = \emptyset$ .
2:  for (each solution  $\pi$  in  $\mathcal{A}$ ) do
3:     $s = 0$ .
4:    if ( $\text{rand}(0,1) \leq \gamma \cdot F_c / F_M$ ) then
5:      while ( $s \leq \varepsilon \cdot m$ ) do
6:        Apply a mutation operation to  $\pi$  to generate its neighborhood solution  $\pi'$ .
7:        if ( $\pi'$  dominates  $\pi$ ) then
8:           $\pi = \pi'$ .
9:        end if
10:        $\omega = \omega \cup \{\pi'\}$ .
11:        $s = s + 1$ .
12:     end while
13:   end if
14: end for
15: update  $\mathcal{A}$  by using  $\omega$ .
```

2.2.5. Evaluation of Obtained Solutions

In the constructed multi-objective optimization model, the penalty of boarding-time adjustment and the penalty of detour in objective function f_1 , as well as in objective function f_2 , can all be computed directly and exactly. However, the passenger waiting penalty cost in objective function f_1 involves stochastic travel time and takes the form of an expectation, which is difficult to evaluate exactly. To address this, the present study employs a stochastic simulation approach, namely Monte Carlo simulation, to approximate this expected value. Specifically, multiple travel-time samples are generated by independently drawing from the probability distributions of travel time between locations, and the passenger waiting penalty cost corresponding to each sample is computed, with the sample mean adopted as an approximation of the expected value. In theory, under ideal conditions with unlimited computational resources, this approach can recover the true value of

the expectation; however, since computational resources are invariably limited, 100 samples are used in this study to approximate the expected value. To ensure a fair comparison among different algorithms, the same set of Monte Carlo travel-time samples (generated using an identical random seed) is used when evaluating all candidate solutions and benchmark algorithms. The specific procedure for evaluating solution π is as follows:

Step 1: Set the sample size $Q = 100$, and obtain Q samples at random. Each sample ρ_q contains the travel time among points, where $q \in N_+ \cup \emptyset$;

Step 2: Compute the total penalty cost $v_q(\pi)$ of each sample ρ_q , where $q \in N_+ \cup \emptyset$;

Step 3: As shown in Equation (24), the mean value $\bar{v}(\pi)$ computed over Q samples is adopted as an approximation of the passenger waiting penalty:

$$\bar{v}(\pi) = \frac{\sum_{q \in N_+ \cup \emptyset} v_q(\pi)}{Q} \quad (24)$$

2.2.6. Procedure of the Proposed Algorithm

The pseudo-code of MOEA-ILSM is shown in Algorithm 3. First, a set of individuals is generated as the initial population. Then, the algorithm performs an iteration over 4 to 6 lines until a termination criterion is reached. In each iteration, the genetic operations process and iterated local search process are executed. Finally, the non-dominated solutions in the external archive are output.

To balance convergence and diversity, MOEA-ILSM integrates multiple complementary strategies. Specifically, the external archive preserves elite non-dominated solutions throughout the search process, while the iterated local search intensifies the search around promising regions. In addition, population diversity is maintained through the hybrid initialization strategy, which combines clustering-based heuristic solutions with randomly generated solutions, as well as five complementary neighborhood structures and the simulated-annealing-inspired acceptance mechanism employed during local search. Together, these mechanisms enhance both convergence performance and population diversity throughout the evolutionary search process.

Algorithm 3. MOEA-ILSM.

Input: Algorithm parameters N , p_m , γ , and ε .

Output: External archive $\mathcal{A}$.

- 1: Initialize population $\mathcal{P}$ with N individuals and evaluate them.
 - 2: Initialize external archive $\mathcal{A}$.
 - 3: **while** (a given termination condition is not met) **do**
 - 4: Perform genetic operations on the population.
 - 5: Update $\mathcal{A}$.
 - 6: Apply iterated local search to the external archive.
 - 7: **end while**
-

3. Results

To evaluate the effectiveness of MOEA-ILSM in addressing the proposed problem, three representative multi-objective evolutionary algorithms are selected for comparison: NSGA-II, MOEA/D, and MOABC. NSGA-II and MOEA/D are well-known algorithms in the field of multi-objective optimization. Their effectiveness has been extensively verified in various applications, such as disassembly optimization [39–41], vehicle routing [42–44],

and flow shop scheduling [45–47]. MOABC is a recently proposed algorithm for tackling multi-objective parallel machine scheduling problem.

All algorithms are implemented in MATLAB R2021b (MathWorks, Natick, MA, USA), and the experiments are conducted on a computer equipped with 16 GB of RAM and an Intel Core i5-9300H processor. To ensure a fair comparison between the proposed algorithm and the baseline algorithms, all algorithms adopt the number of fitness evaluations as the termination criterion. The maximum number of evaluations is set to the product of three factors: the total number of requests, the proportion of requests accepting boarding time recommendation, and 500.

3.1. Test Instance Construction

In this study, the proposed algorithm is evaluated using the DiDi Chuxing dataset and the road network of Chengdu, China. Specifically, requests collected on 16 November 2016 are selected. As presented in Table 2, the instances are classified into four categories: S1, M1, M2, and L1, and each category contains four instances. The categorization is determined by time duration, the ratio of requests accepting BT recommendation, the number of requests (m), and the number of vehicles (n). The notation ‘X-Y’ denotes the Y-th instance of category X, where $X \in \{S1, M1, M2, L1\}$ specifies the type, and $Y \in \{1, 2, 3, 4\}$ indicates the instance.

As illustrated in Figure 5, the road network of a subregion in Chengdu is imported into SUMO [48] to compute driving distances between points. We gathered requests with both pick-up and drop-off points located within the blue-marked area. In each instance, the requests that accept the BT recommendation are randomly selected according to the ratio in Table 2. For each request, the initial boarding time is set to the order time. In addition, the number of passengers per request is randomly assigned, with the proportions of requests containing one, two, three, and four passengers being 40%, 30%, 20%, and 10%, respectively. The initial points of vehicles are generated randomly, and all vehicles are homogeneous, with a capacity of four passengers. The penalty coefficients for vehicle tardiness, request boarding time adjustment, and vehicle detour are set as $\zeta_1=5$, $\zeta_2=2$, and $\zeta_3=1$, respectively. The maximum allowable deviation of the boarding time, $\hat{\psi}$, is set to 15.

Consistent with existing studies [49], the travel time $t_{\alpha,\beta}$ follows a left-truncated normal distribution with mean $l_{\alpha,\beta}/0.5$, where α and β denote any two points and the constant 0.5 km/min corresponds to a travel speed of 30 km/h. The standard deviation $\text{std}(t_{\alpha,\beta})$ is defined as $U[0.1, 0.6] \cdot l_{\alpha,\beta}/0.5$, where U denotes a uniformly distributed random variable. This setting corresponds to a coefficient of variation ranging from 10% to 60%, covering different levels of traffic uncertainty commonly encountered in urban transportation systems.



Figure 5. Road network of a subarea in Chengdu, China. The blue box indicates the area in which both the pick-up and drop-off points of the collected requests are located.

Table 2. Description of instances.

Instance	Time Slot	Ratio of Requests Accepting BT Recommendation	m	n
S1-1	08:20–08:30	20%	30	15
S1-2	08:10–08:20	20%	34	15
S1-3	08:00–08:10	20%	36	15
S1-4	08:30–08:40	20%	39	15
M1-1	10:10–10:30	40%	60	30
M1-2	15:40–16:00	40%	62	30
M1-3	09:00–09:20	40%	64	30
M1-4	13:30–13:50	40%	66	30
M2-1	13:00–13:20	60%	68	30
M2-2	15:00–15:20	60%	68	30
M2-3	09:40–10:00	60%	69	30
M2-4	10:40–11:00	60%	69	30
L1-1	19:00–19:40	80%	103	50
L1-2	12:00–12:40	80%	106	50
L1-3	16:00–16:40	80%	109	50
L1-4	18:30–19:10	80%	109	50

3.2. Performance Metrics

This study employs three performance evaluation metrics: IGD [50], hypervolume [51], and the C-metric [52]. The definitions of them are provided below.

- (1) IGD measures the approximation and distribution quality of a solution set by computing the average distance from the Pareto-optimal set, denoted by X^* , to the obtained solution set, denoted by X . Its calculation is expressed as follows:

$$IGD(X^*, X) = \frac{\sum_{x \in X^*} \text{dist}(x, X)}{|X^*|} \quad (25)$$

where $\text{dist}(x, X)$ denotes the Euclidean distance between a solution x in X^* and the nearest solution in X . In fact, X^* cannot be obtained for the studied problem. Therefore, we combine all solutions generated by all algorithms and treat the non-dominated ones as X^* . In addition, before calculating the IGD value, all objective function values are normalized to $[0, 1]$. A smaller IGD value corresponds to a better solution set X .

- (2) The C-metric is employed to measure the dominance relationship between two solution sets. Let Y and Z denote two solution sets. $C(Y, Z)$ is defined as the fraction of solutions in Z that are dominated by at least one solution in Y , and it is calculated as follows:

$$C(Y, Z) = \frac{|\{z \in Z \mid \exists y \in Y : y \prec z\}|}{|Z|} \quad (26)$$

where $|Z|$ represents the total number of solutions in Z . A bigger value of $C(Y, Z)$ indicates that Y outperforms Z .

3.3. Parameter Setting of MOEA-ILSM

To investigate the impact of parameters on the performance of MOEA-ILSM, the Taguchi design of experiment method [53] is employed to analyze its parameters. Each parameter is assigned four levels: $N \in \{40, 50, 60, 70\}$, $p_m \in \{0.05, 0.10, 0.15, 0.20\}$, $\gamma \in \{0.40, 0.60, 0.80, 1.00\}$, and $\varepsilon \in \{0.80, 1.00, 1.20, 1.40\}$. Accordingly, an orthogonal array L16(4⁴) is adopted to generate 16 distinct parameter combinations, as detailed in Table 3.

For each combination, MOEA-ILSM is executed 20 times on instance M1-1. The performance is assessed using the IGD, and the average IGD (named AIGD) values across 20

independent runs are reported in Table 3. In addition, the significance rank of parameters is presented in Table 4, where “Delta” represents the maximum difference in AIGD across the four levels of a parameter, and a larger Delta value indicates a greater influence of that parameter on the algorithmic performance.

According to the results presented in Table 3, γ exhibits the greatest impact on the performance of MOEA-ILSM, with p_m , N , and ε ranking second, third, and fourth, respectively. Figure 6 illustrates the influence trends of each parameter. Based on the results presented in Table 4 and Figure 6, a parameter combination yielding superior performance can be identified as follows: $N = 50$, $p_m = 0.10$, $\gamma = 0.60$, and $\varepsilon = 1.20$. Accordingly, this parameter combination is adopted for MOEA-ILSM in all subsequent experiments.

Table 3. Orthogonal table and AIGD values of MOEA-ILSM.

No.	N	p_m	γ	ε	AIGD
1	40	0.05	0.40	0.80	0.1957
2	40	0.10	0.60	1.00	0.1876
3	40	0.15	0.80	1.20	0.1993
4	40	0.20	1.00	1.40	0.2032
5	50	0.05	0.60	1.20	0.1828
6	50	0.10	0.40	1.40	0.1868
7	50	0.15	1.00	0.80	0.2041
8	50	0.20	0.80	1.00	0.1950
9	60	0.05	0.80	1.40	0.1995
10	60	0.10	1.00	1.20	0.2028
11	60	0.15	0.40	1.00	0.2050
12	60	0.20	0.60	0.80	0.1998
13	70	0.05	1.00	1.00	0.1996
14	70	0.10	0.80	0.80	0.1875
15	70	0.15	0.60	1.40	0.1962
16	70	0.20	0.40	1.20	0.1976

Table 4. Influence trend and rank of MOEA-ILSM.

Level	N	p_m	γ	ε
1	0.1965	0.1944	0.1963	0.1968
2	0.1922	0.1912	0.1916	0.1968
3	0.2018	0.2012	0.1953	0.1956
4	0.1952	0.1989	0.2024	0.1964
Delta	0.0096	0.0100	0.0108	0.0012
Rank	3	2	1	4

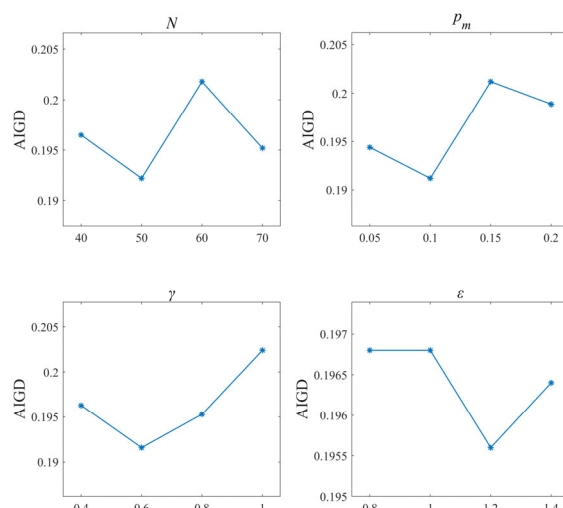


Figure 6. Influence trend of parameters.

According to papers [49,54,55], the parameter settings for NSGA-II, MOEA/D, and MOABC are given in Table 5.

Table 5. Parameter settings of comparative algorithms.

Algorithm	Parameter Setting
NSGA-II	population size: 100, mutation probability: 0.8
MOEA/D	population size: 100, mutation probability: 1.0, neighborhood size: 20
MOABC	population size: 30, maximum trial number: 6

3.4. Effectiveness of Heuristic Method in Initialization

The heuristic method is designed to generate high-quality initial individuals. To validate its effectiveness, the proposed heuristic method is compared with the random initialization, where initial individuals are generated randomly.

Taking instance M1-1 as a representative case, a population initialization experiment is conducted. The threshold for the number of requests in each cluster is set to $\zeta = 3$, the distance-related coefficient is set to $\iota = 0.3$, and the angle threshold between directed line segments is set to $\theta = 90^\circ$. The established model seeks to minimize f_1 and f_2 , i.e., the total penalty and total driving distance. Figure 7 displays the normalized objective values of the initial population, where (a) corresponds to individuals generated by the proposed heuristic method and (b) corresponds to those produced by random initialization. In both subfigures, each row represents one solution, and the two columns correspond to the two normalized objectives. Most individuals generated by the heuristic method exhibit lower objective values compared to those produced by random initialization. It is evident that the quality of the individuals generated by the heuristic method is superior to that generated by random initialization.

To provide a more intuitive illustration of the results, the two boxplots in Figure 8 present the distributions of objective values for the initial populations generated by the proposed heuristic method and by random initialization. The results indicate that the median total penalty obtained by the heuristic method is 488, which is significantly lower than 755 obtained by random initialization. Similarly, the median total travel distance produced by the heuristic method is 118, which is also lower than 146 produced by random initialization. These findings demonstrate that the heuristic initialization method can effectively improve the quality of the initial population.

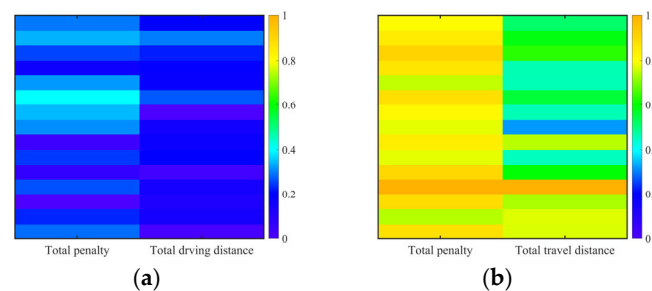


Figure 7. The normalized objective values under different methods: (a) heuristic method and (b) random initialization.

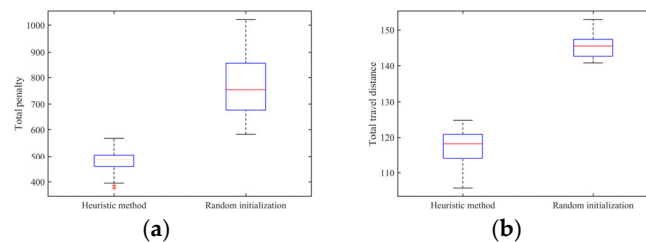


Figure 8. The comparison of the objective values: (a) total operation cost and (b) total penalty cost.

3.5. Effectiveness of Iterated Local Search Method

MOEA-ILSM utilizes an external archive to store the non-dominated solutions obtained during the optimization process. To further enhance the algorithm's performance, an iterated local search method (ILSM) is designed, which intensively explores the candidate solutions in $\mathcal{A}$, thereby strengthening the local search capability of MOEA-ILSM. To evaluate the contribution of ILSM, this paper removes ILSM from the original algorithm and constructs a comparison algorithm without this method, denoted as MOEA. Subsequently, comparative experiments are conducted on experimental instances to verify the necessity of ILSM in improving the algorithm's solution performance.

This study adopts the IGD as the evaluation criterion to comprehensively assess the optimization effectiveness of the algorithms. Table 6 presents the comparative experimental results of the two algorithms. The results demonstrate that, although MOEA-ILSM exhibits a slightly higher average variance in the IGD, its IGD values are substantially lower than those of MOEA across all instances. This indicates that MOEA-ILSM achieves superior performance in terms of solution set convergence, distribution uniformity, and coverage range. To further examine the statistical significance of the differences between MOEA-ILSM and MOEA, a t -test with 38 degrees of freedom at a significance level of 0.05 is employed [56]. The symbols “+”, “~”, and “−” denote that MOEA-ILSM performs significantly better than, statistically comparable to, or significantly worse than the reference algorithm, respectively, with the corresponding results reported in the “ t -test” column of Table 4. The t -test further confirms that the performance improvements achieved by MOEA-ILSM over MOEA are statistically significant. In summary, the comparative experimental results demonstrate that the incorporation of ILSM effectively enhances the algorithm's capability to search for high-quality solutions, thereby further improving the overall optimization performance of MOEA-ILSM.

Table 6. Comparisons of MOEA-ILSM and MOEA in terms of the IGD.

Instance	MOEA-ILSM		MOEA		<i>t</i> -Test
	Mean	Variance	Mean	Variance	
S1-1	0.1003	0.0015	0.1389	0.0010	+
S1-2	0.0955	0.0008	0.1690	0.0010	+
S1-3	0.0785	0.0005	0.1165	0.0006	+
S1-4	0.0648	0.0011	0.1705	0.0006	+
M1-1	0.0527	0.0005	0.1302	0.0004	+
M1-2	0.0867	0.0004	0.1287	0.0003	+
M1-3	0.0783	0.0014	0.1825	0.0004	+
M1-4	0.0565	0.0003	0.1294	0.0006	+
M2-1	0.0885	0.0014	0.2199	0.0016	+
M2-2	0.1073	0.0022	0.3013	0.0010	+
M2-3	0.0487	0.0006	0.1452	0.0010	+
M2-4	0.0939	0.0009	0.2492	0.0019	+
L1-1	0.1344	0.0010	0.1847	0.0003	+
L1-2	0.1506	0.0004	0.1752	0.0010	+
L1-3	0.0795	0.0002	0.1535	0.0005	+
L1-4	0.0987	0.0008	0.1968	0.0009	+
Average	0.0884	0.0009	0.1745	0.0008	

3.6. Comparisons of MOEA-ILSM with the Three Algorithms

In this section, to validate the superiority of MOEA-ILSM, it is compared against three representative algorithms, namely NSGA-II, MOEA/D, and MOABC. For each benchmark instance, all algorithms are executed independently for 20 runs, and the mean and variance of the evaluation indicators are recorded as the basis for performance assessment.

Table 7 presents the comparative results of the four methods in terms of the C-metric. In this table, MOEA-ILSM, NSGA-II, MOEA/D, and MOABC are denoted by the symbols “EA”, “GA”, “EAD”, and “ABC”, respectively. A comparison between MOEA-ILSM and NSGA-II reveals that, in 9 out of 16 instances, the solutions obtained by MOEA-ILSM completely dominate all solutions produced by NSGA-II, i.e., $C(EA, GA) = 1$; meanwhile, in 15 instances, the solutions of NSGA-II fail to dominate any solution found by MOEA-ILSM, i.e., $C(GA, EA) = 0$. When comparing MOEA-ILSM with MOEA/D, it is observed that the solutions obtained by MOEA-ILSM completely dominate all solutions of MOEA/D in 5 instances, i.e., $C(EA, EAD) = 1$, while in 15 instances, the solutions found by MOEA/D are unable to dominate any solution of MOEA-ILSM, i.e., $C(EAD, EA) = 0$. Similarly, in 14 instances, the solutions obtained by MOEA-ILSM dominate all solutions produced by MOABC, and in 14 instances, the solutions of MOABC fail to outperform any solution obtained by MOEA-ILSM. Based on the foregoing analysis, it can be concluded that MOEA-ILSM is capable of identifying a greater number of dominating solutions than its counterparts, demonstrating a clear advantage over the three comparison algorithms with respect to the C-metric.

Table 7. Experimental results of four algorithms in terms of the C-metric.

Instance	C(EA, GA)	C(GA, EA)	C(EA, EAD)	C(EAD, EA)	C(EA, ABC)	C(ABC, EA)
S1-1	0.9425	0.0065	0.9200	0.0132	0.9016	0.0596
S1-2	0.9917	0.0000	1.0000	0.0000	1.0000	0.0000
S1-3	0.9732	0.0000	0.9714	0.0000	1.0000	0.0000
S1-4	0.9833	0.0000	1.0000	0.0000	0.9917	0.0033
M1-1	1.0000	0.0000	0.9238	0.0000	1.0000	0.0000
M1-2	1.0000	0.0000	0.8784	0.0000	1.0000	0.0000
M1-3	0.9967	0.0000	1.0000	0.0000	1.0000	0.0000
M1-4	1.0000	0.0000	0.9739	0.0000	1.0000	0.0000
M2-1	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
M2-2	1.0000	0.0000	0.9967	0.0000	1.0000	0.0000
M2-3	1.0000	0.0000	0.9718	0.0000	1.0000	0.0000
M2-4	0.9983	0.0000	1.0000	0.0000	1.0000	0.0000
L1-1	1.0000	0.0000	0.6288	0.0000	1.0000	0.0000
L1-2	0.9967	0.0000	0.5908	0.0000	1.0000	0.0000
L1-3	1.0000	0.0000	0.7148	0.0000	1.0000	0.0000
L1-4	1.0000	0.0000	0.6871	0.0000	1.0000	0.0000
Average	0.9927	0.0004	0.8911	0.0008	0.9933	0.0039

Table 8 presents the comparative results of MOEA-ILSM and its counterpart algorithms in terms of the IGD. As can be observed from Table 8, MOEA-ILSM achieves the lowest IGD values among all algorithms across every benchmark instance, demonstrating superior solution quality relative to all other comparison algorithms. With respect to the mean values, the averages of MOEA-ILSM, NSGA-II, MOEA/D, and MOABC are 0.0884, 0.2367, 0.3213, and 0.3568, respectively; with respect to the variance, the corresponding averages are 0.0009, 0.0008, 0.0016, and 0.0038, respectively. Although the average variance of MOEA-ILSM is marginally higher than that of NSGA-II, its average mean value is substantially lower. In comparison with MOEA/D and MOABC, MOEA-ILSM attains lower average values in both mean and variance. In addition, Figure 9 presents the box plots of the IGD for MOEA-ILSM and its comparison algorithms across four instances. As can be observed, MOEA-ILSM achieves a substantially lower median IGD value than all comparison algorithms across every instance. Therefore, it can be concluded that MOEA-ILSM significantly outperforms the other three comparison algorithms in terms of the IGD on the problem under investigation.

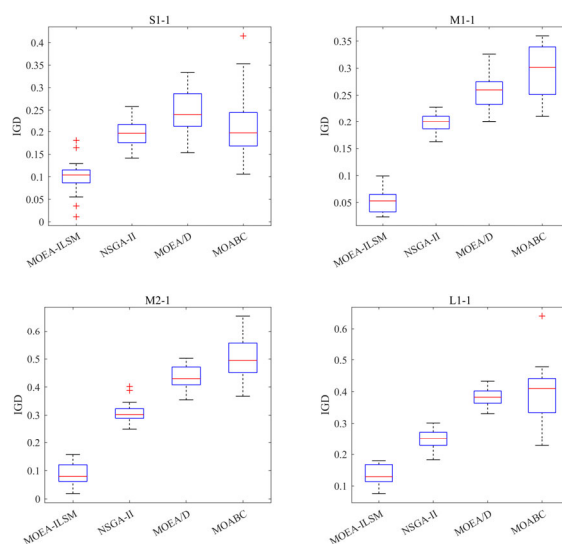
**Figure 9.** Boxplots of the IGD in four instances.

Table 8. Experimental results of four algorithms in terms of the IGD.

Instance	MOEA-ILSM		NSGA-II		MOEA/D		MOABC	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
S1-1	0.0929	0.0012	0.1970	0.0010	0.2561	0.0035	0.2438	0.0071
S1-2	0.1380	0.0032	0.2750	0.0014	0.3425	0.0015	0.3131	0.0031
S1-3	0.0595	0.0004	0.1733	0.0007	0.2250	0.0015	0.2127	0.0033
S1-4	0.0776	0.0011	0.2368	0.0006	0.2902	0.0013	0.2581	0.0022
M1-1	0.0684	0.0005	0.2736	0.0014	0.4309	0.0019	0.4466	0.0040
M1-2	0.0796	0.0010	0.2148	0.0010	0.3292	0.0016	0.4255	0.0044
M1-3	0.0435	0.0004	0.1288	0.0003	0.2234	0.0006	0.2063	0.0008
M1-4	0.1067	0.0019	0.3036	0.0009	0.4255	0.0018	0.4538	0.0063
M2-1	0.0944	0.0020	0.3198	0.0013	0.4631	0.0021	0.5740	0.0068
M2-2	0.0591	0.0013	0.2180	0.0009	0.2746	0.0013	0.3628	0.0012
M2-3	0.0545	0.0005	0.2048	0.0005	0.2817	0.0007	0.3715	0.0015
M2-4	0.0669	0.0004	0.1708	0.0007	0.2308	0.0011	0.3079	0.0012
L1-1	0.1190	0.0004	0.2112	0.0007	0.3479	0.0007	0.3485	0.0091
L1-2	0.1155	0.0006	0.2028	0.0007	0.3484	0.0012	0.4278	0.0080
L1-3	0.0843	0.0004	0.2344	0.0010	0.3583	0.0012	0.3918	0.0142
L1-4	0.0760	0.0002	0.1667	0.0004	0.2346	0.0003	0.3256	0.0022
Average	0.0835	0.0010	0.2207	0.0008	0.3164	0.0014	0.3544	0.0047

Furthermore, to examine the statistical significance of the performance differences between MOEA-ILSM and each comparison algorithm, a *t*-test is conducted at a significance level of 0.05 with 38 degrees of freedom. Table 9 presents the statistical test results based on the C-metric and the IGD. As can be observed from the results, MOEA-ILSM significantly outperforms NSGA-II, MOEA/D, and MOABC across all instances under two metrics. Therefore, it can be concluded that MOEA-ILSM demonstrates stronger optimization capability in solving the problem under investigation.

Table 9. *t*-test results of ILS-MOEA against its peers in terms of the IGD and the C-metric.

Instance	MOEA-ILSM vs.					
	NSGA-II		MOEA/D		MOABC	
	C	IGD	C	IGD	C	IGD
S1-1	+	+	+	+	+	+
S1-2	+	+	+	+	+	+
S1-3	+	+	+	+	+	+
S1-4	+	+	+	+	+	+
M1-1	+	+	+	+	+	+
M1-2	+	+	+	+	+	+
M1-3	+	+	+	+	+	+
M1-4	+	+	+	+	+	+
M2-1	+	+	+	+	+	+
M2-2	+	+	+	+	+	+
M2-3	+	+	+	+	+	+
M2-4	+	+	+	+	+	+
L1-1	+	+	+	+	+	+
L1-2	+	+	+	+	+	+
L1-3	+	+	+	+	+	+
L1-4	+	+	+	+	+	+

To assess the computational efficiency of MOEA-ILSM and the three benchmark algorithms on the considered problem, this study measures their computation time across the largest instances (L1). As shown in Table 10, NSGA-II consistently requires the shortest runtime for all instances, with MOEA-ILSM ranking second, followed by MOEA/D and MOABC. Although MOEA-ILSM incurs a slightly higher computational cost than NSGA-II, the difference is negligible and acceptable for the problem considered. Furthermore, this algorithm shows strong potential for future deployment in real-time dispatching systems.

Table 10. Computation time of MOEA-ILSM and the three benchmark algorithms (measured in seconds).

Instance	MOEA-ILSM	NSGA-II	MOEA/D	MOABC
L1-1	60.8420	52.1761	72.1841	87.5317
L1-2	66.2282	53.7417	77.3457	89.5321
L1-3	71.8021	55.2307	82.6659	97.7453
L1-4	71.9345	55.4293	81.0772	97.7936

Figure 10 visualizes the distribution of non-dominated solution sets generated by each algorithm across four benchmark instances of varying scales. As can be observed, the non-dominated solutions obtained by MOEA-ILSM are closer to the lower-left region of the figure compared to those of the other comparison algorithms. This indicates that MOEA-ILSM exhibits superior convergence performance in solving the problem under investigation.

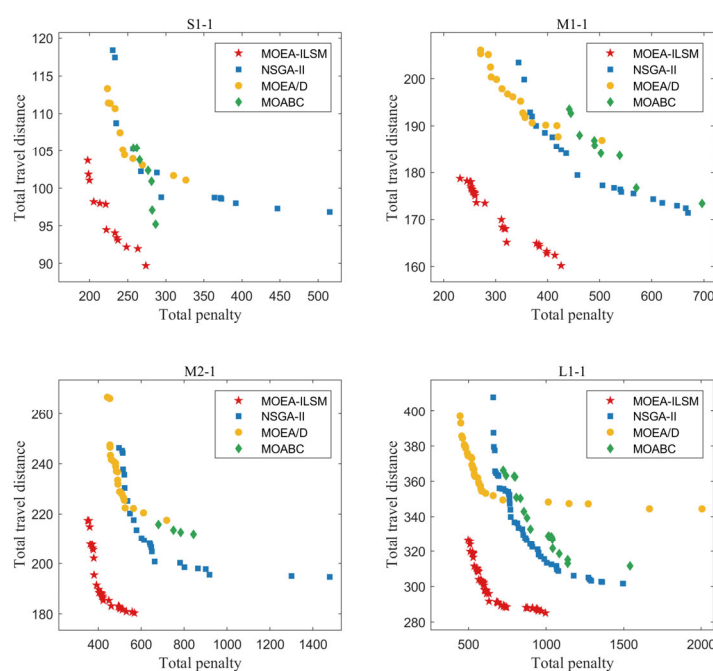


Figure 10. Non-dominated solutions obtained in four instances.

3.7. Experiments on an Additional Test Region

To further evaluate the generalizability of the proposed MOEA-ILSM in different traffic environments, an additional test region in Chengdu is introduced in this subsection. Compared to the original experimental region, the new region is characterized by a higher traffic congestion level. The purpose of this experiment is to investigate whether MOEA-ILSM can maintain its optimization performance when applied to a different urban environment. Figure 11 illustrates the geographical distribution of the additional test region,

which is highlighted by a red box. Similar to the original study area, the road network is extracted from OpenStreetMap, and the travel requests are generated according to the historical demand distribution observed in the region.

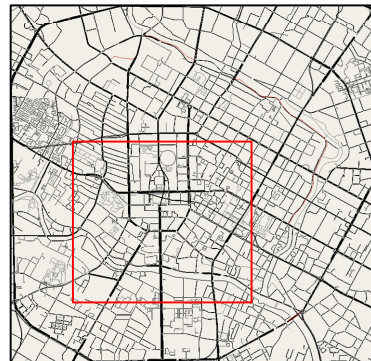


Figure 11. Road network of the additional test region, where the red box indicates the selected study area.

To simulate a congested traffic environment, the mean value of vehicle travel speed is reduced from the setting adopted in the original experiments to 18 km/h. The travel speed on each road segment is assumed to follow a truncated normal distribution, while the corresponding travel time is calculated according to the length of the road. Other parameter settings remain identical to those used in the previous experiments. Considering that the objective of this experiment is to verify the robustness of the algorithms under a different traffic environment rather than conducting a comprehensive benchmark study, four representative instances are generated from the additional region. The characteristics of these instances are summarized in Table 11.

Table 11. Experimental instances in the additional test region.

Instance	Time Slot	Ratio of Requests Accepting BT Recommendation	m	n
S2-1	08:40–08:50	20%	30	15
M3-1	11:00–11:20	40%	60	30
M4-1	11:20–11:40	60%	68	30
L2-1	10:00–10:40	80%	103	50

To assess the optimization performance of the algorithms, MOEA-ILSM is compared with NSGA-II, MOEA/D, and MOABC using the IGD. Each algorithm is independently executed 20 times for every instance, and the mean and variance of the IGD values are recorded. The experimental results are reported in Table 12.

Table 12. Experimental results of four algorithms regarding the IGD on the additional test region.

Instance	MOEA-ILSM		NSGA-II		MOEA/D		MOABC	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
S2-1	0.1032	0.0027	0.3058	0.0008	0.3778	0.0040	0.3093	0.0104
M3-1	0.0575	0.0009	0.2211	0.0014	0.2861	0.0013	0.3053	0.0025
M4-1	0.0742	0.0014	0.3195	0.0014	0.4513	0.0014	0.5159	0.0037
L2-1	0.0833	0.0010	0.4154	0.0014	0.5550	0.0025	0.7234	0.0065
Average	0.0796	0.0015	0.3155	0.0013	0.4176	0.0023	0.4635	0.0058

According to the results presented in Table 11, MOEA-ILSM consistently achieves the lowest IGD values across all benchmark instances. The average IGD value obtained by MOEA-ILSM is 0.0796, which is significantly lower than those achieved by NSGA-II,

MOEA/D, and MOABC, whose average IGD values are 0.3155, 0.4176, and 0.4635, respectively. These results indicate that the solution sets generated by MOEA-ILSM exhibit superior convergence and distribution quality even under more congested traffic conditions. Overall, the experimental results obtained on the additional test region confirm that MOEA-ILSM possesses strong robustness and generalizability. Therefore, the proposed algorithm can be effectively applied to robotaxi dispatching problems under different urban traffic conditions.

3.8. Analysis of the Superiority of Boarding-Time Recommendations

This study innovatively incorporates a boarding-time recommendation mechanism and proposes the stochastic robotaxi dispatch problem with boarding-time recommendation (SRD-BTR), along with a corresponding multi-objective stochastic programming model. To verify the effectiveness of this mechanism, a comparative variant model is constructed by setting all passengers to reject the boarding-time recommendation, thereby yielding a model corresponding to the robotaxi dispatch problem without the boarding-time recommendation mechanism. To further analyze the impact of this mechanism on system performance, ablation experiments are conducted on four benchmark instances of varying scales, and MOEA-ILSM is employed to solve both models. For each instance, the algorithm is executed independently 20 times for each model, and each run yields a non-dominated solution set. The total expected passenger waiting time and the order carpooling rate corresponding to each solution set are computed, and the averages over 20 runs are adopted as the evaluation indicators. Figure 10 illustrates the experimental results of the two models across instances of different scales.

As can be observed from Figure 12, across all instances, the model incorporating the boarding-time recommendation mechanism consistently achieves substantially lower expected passenger waiting times than the comparison model without this mechanism. Specifically, for the small-scale instance S1-1, the total expected waiting time of the model with the recommendation mechanism is 100.7 min, compared to 103.7 min for the comparison model, representing a reduction of approximately 2.89%. For the medium-scale instance M1-1, the total expected waiting time decreases from 99.0 min to 88.7 min, a reduction of approximately 10.40%. For instance M2-1, this metric decreases from 112.5 min to 88.3 min, a reduction of 21.51%, which is the most pronounced improvement among the four instances. For the larger-scale instance L1-1, the total expected waiting time decreases from 125.9 min to 83.8 min, a reduction of approximately 33.44%. Furthermore, as the instance scale expands, the number of available vehicles increases accordingly, which enhances the likelihood of passengers being served by nearby vehicles; consequently, the total expected waiting time does not increase monotonically with scale, a pattern observed in both models. These results indicate that by appropriately recommending boarding times for certain passengers, vehicles are afforded greater flexibility in adjusting their routes, thereby reducing the expected passenger waiting time.

With respect to the request carpooling rate shown in Figure 10, the model incorporating the boarding-time recommendation mechanism outperforms the comparison model across all benchmark instances. For instance M1-1, the carpooling rate increases from 0.404 to 0.415; for instance L1-1, it rises from 0.377 to 0.402, representing the most substantial improvement; and the same trend is also observed for instances S1-1 and M2-1. This demonstrates that when passengers are permitted to adjust their boarding times within a certain range, the dispatching system can more readily consolidate multiple travel demands into carpooling services, thereby improving the utilization efficiency of vehicle resources.

In summary, the incorporation of a boarding-time recommendation mechanism into the ride-hailing dispatching problem can effectively reduce the expected passenger waiting time and improve the request carpooling rate through moderate adjustments to passenger boarding time. The experimental results thoroughly validate the effectiveness and superiority of this mechanism.

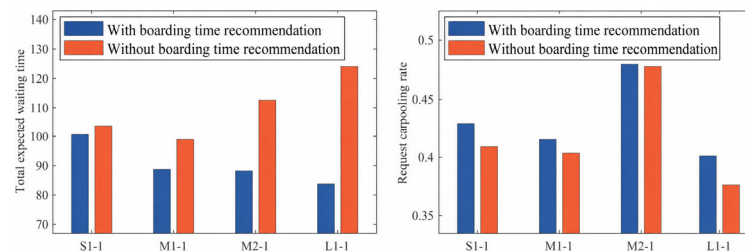


Figure 12. Results of the ablation test.

4. Conclusions

This study investigates a stochastic robotaxi dispatch problem with boarding-time recommendation (SRD-BTR), taking into account the uncertainty of vehicle travel time. A multi-objective mixed-integer programming model is formulated with the objectives of minimizing the total penalty cost associated with passenger travel experience and minimizing the total vehicle travel distance. A multi-objective evolutionary algorithm with an iterated local search method (MOEA-ILSM) is proposed to solve the model. Experimental results demonstrate that the designed heuristic initialization strategy effectively enhances the quality of the initial population, and that the iterated local search significantly strengthens the search capability of the algorithm. Comparative experiments conducted against NSGA-II, MOEA/D, and MOABC show that MOEA-ILSM achieves superior performance in terms of both the IGD and the C-metric. Finally, ablation experiments are carried out to validate the superiority of the proposed model.

The current study evaluates the proposed boarding-time recommendation mechanism in a controlled offline simulation setting, where all requests are assumed to be known in advance to facilitate a clear assessment of system performance. The results demonstrate the potential of the proposed approach to improve ride-sharing efficiency and service quality, providing a foundation for future extensions to more dynamic operational settings. The travel time distribution is also specified under commonly adopted assumptions in simulation-based ride-sharing studies, without calibration to a specific real-world dataset. Future research may extend the proposed framework to dynamic ride-sharing environments using rolling-horizon or event-triggered optimization strategies. It may also consider additional real-world operational factors, such as passenger behavioral responses, communication delays, system disruptions, and large-scale traffic disturbances caused by accidents, road closures, or adverse weather conditions, to further enhance practical applicability.

As the problem scale continues to expand, traditional multi-objective evolutionary algorithms may face challenges in terms of computational efficiency. Future research could further explore novel hybrid solution approaches that integrate deep learning, reinforcement learning, and heuristic optimization algorithms, with the aim of improving both the solution efficiency and stability of algorithms when applied to large-scale robotaxi dispatch problems.

Author Contributions: Conceptualization, Y.Z. and L.Q.; Methodology, Q.X. and L.Q.; Software, Q.X. and L.Q.; Validation, Y.Z., Y.P. and L.Q.; Formal analysis, Y.Z.; Investigation, Y.Z. and L.Q.; Resources, Y.Z.; Data curation, Q.X.; Writing – original draft, Y.Z. and Q.X.; Writing – review &

editing, Y.P. and L.Q.; Visualization, Q.X. and Y.P.; Supervision, L.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Natural Science Foundation of Shandong Province under Grant ZR2024MF140 and in part by the National Natural Science Foundation of China under Grant 61903229.

Institutional Review Board Statement: Not applicable, as this study did not involve humans or animals.

Informed Consent Statement: Not applicable, as this study did not involve humans.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

References

1. Hussain, R.; Zeadally, S. Autonomous cars: Research results, issues, and future challenges. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 1275–1313.
2. Eskandarian, A.; Wu, C.; Sun, C. Research advances and challenges of autonomous and connected ground vehicles. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 683–711.
3. Dai, J.; Wang, X.C.; Ma, W.; Li, R. Future transport vision propensity segments: A latent class analysis of autonomous taxi market. *Transp. Res. Part A Policy Pract.* **2023**, *173*, 103699.
4. Zhang, X.; Sun, H.; Pei, X.; Guan, L.; Wang, Z. Evolution of technology investment and development of robotaxi services. *Transp. Res. Part E Logist. Transp. Rev.* **2024**, *188*, 103615.
5. Dandl, F.; Bogenberger, K. Comparing future autonomous electric taxis with an existing free-floating carsharing system. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 2037–2047.
6. Narayanan, S.; Chaniotakis, E.; Antoniou, C. Shared autonomous vehicle services: A comprehensive review. *Transp. Res. Part C Emerg. Technol.* **2020**, *111*, 255–293.
7. Zeng, W.; Wu, M.; Chen, P.; Cao, Z.; Xie, S. Review of shared online hailing and autonomous taxi services. *Transp. B Transp. Dyn.* **2023**, *11*, 486–509.
8. Hu, L.; Dong, J. An artificial-neural-network-based model for real-time dispatching of electric autonomous taxis. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 1519–1528.
9. Li, J.; Yu, C.; Shen, Z.; Su, Z.; Ma, W. A survey on urban traffic control under mixed traffic environment with connected automated vehicles. *Transp. Res. Part C Emerg. Technol.* **2023**, *154*, 104258.
10. Mańdziuk, J. New shades of the vehicle routing problem: Emerging problem formulations and computational intelligence solution methods. *IEEE Trans. Emerg. Top. Comput. Intell.* **2019**, *3*, 230–244.
11. Seo, T.; Asakura, Y. Multi-objective linear optimization problem for strategic planning of shared autonomous vehicle operation and infrastructure design. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 3816–3828.
12. Miao, F.; Han, S.; Lin, S.; Wang, Q.; Stankovic, J.A.; Hendawi, A.; Zhang, D.; He, T.; Pappas, G.J. Data-driven robust taxi dispatch under demand uncertainties. *IEEE Trans. Control Syst. Technol.* **2019**, *27*, 175–191.
13. Kim, B.; Kim, J.; Huh, S.; You, S.; Yang, I. Multi-objective predictive taxi dispatch via network flow optimization. *IEEE Access* **2020**, *8*, 21437–21452.
14. Kim, B.; Huh, S. Discretization-free particle-based taxi dispatch methods with network flow decomposition. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 17756–17768.
15. Duan, L.; Wei, Y.; Zhang, J.; Xia, Y. Centralized and decentralized autonomous dispatching strategy for dynamic autonomous taxi operation in hybrid request mode. *Transp. Res. Part C Emerg. Technol.* **2020**, *111*, 397–420.
16. Chen, Z.G.; Zhan, Z.H.; Kwong, S.; Zhang, J. Evolutionary computation for intelligent transportation in smart cities: A survey [review article]. *IEEE Comput. Intell. Mag.* **2022**, *17*, 83–102.
17. Zhu, M.; Liu, X.Y.; Wang, X. An online ride-sharing path-planning strategy for public vehicle systems. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 616–627.
18. Gong, Y.J.; Liu, Y.W.; Lin, Y.; Chen, W.N.; Zhang, J. Real-time taxi-passenger matching using a differential evolutionary fuzzy controller. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 2712–2725.

19. Rivière, B.; Chung, S.J. H-TD2: Hybrid temporal difference learning for adaptive urban taxi dispatch. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 10935–10944.
20. Liu, Z.; Li, J.; Wu, K. Context-aware taxi dispatching at city-scale using deep reinforcement learning. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 1996–2009.
21. Santi, P.; Resta, G.; Szell, M.; Sobolevsky, S.; Strogatz, S.H.; Ratti, C. Quantifying the benefits of vehicle pooling with shareability networks. *Proc. Natl. Acad. Sci. USA* **2014**, *111*, 13290–13294.
22. Fielbaum, A.; Bai, X.; Alonso-Mora, J. On-demand ridesharing with optimized pick-up and drop-off walking locations. *Transp. Res. Part C Emerg. Technol.* **2021**, *126*, 103061.
23. Alisoltani, N.; Leclercq, L.; Zargayouna, M. Can dynamic ride-sharing reduce traffic congestion? *Transp. Res. Part B Methodol.* **2021**, *145*, 212–246.
24. Ke, J.; Yang, H.; Zheng, Z. On ride-pooling and traffic congestion. *Transp. Res. Part B Methodol.* **2020**, *142*, 213–231.
25. Huang, S.C.; Jiau, M.K.; Liu, Y.P. An ant path-oriented carpooling allocation approach to optimize the carpool service problem with time windows. *IEEE Syst. J.* **2019**, *13*, 994–1005.
26. Li, X.; Gao, J.; Wang, C.; Huang, X.; Nie, Y. Ride-sharing matching under travel time uncertainty through data-driven robust optimization. *IEEE Access* **2022**, *10*, 116931–116941.
27. Huang, S.C.; Lin, J.J.; Jiau, M.K. Global and local Pareto optimality in coevolution for solving carpool service problem with time windows. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 934–946.
28. Guo, Y.; Zhang, Y.; Boulaksil, Y. Real-time ride-sharing framework with dynamic timeframe and anticipation-based migration. *Eur. J. Oper. Res.* **2021**, *288*, 810–828.
29. Liang, X.; Correia, G.H.d.A.; An, K.; van Arem, B. Automated taxis' dial-a-ride problem with ride-sharing considering congestion-based dynamic travel time. *Transp. Res. Part C Emerg. Technol.* **2020**, *112*, 260–281.
30. Zhang, C.; Zhu, F.; Wang, X.; Sun, L.; Tang, H.; Lv, Y. Taxi demand prediction using parallel multi-task learning model. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 794–803.
31. Li, M.; Qi, L.; Luan, W.; Talukder, Q.T.A.; Guo, X. Optimization of robotaxi dispatch with pick-up/drop-off-point and boarding-time recommendation. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 22524–22537.
32. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197.
33. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731.
34. Akbari, R.; Hedayatzadeh, R.; Ziarati, K.; Hassanizadeh, B. A multi-objective artificial bee colony algorithm. *Swarm Evol. Comput.* **2012**, *2*, 39–52.
35. Vincent, F.Y.; Zegeye, M.M.; Gebeyehu, S.G.; Indrakarna, P.A.; Jodiawan, P. The multi-depot general share-a-ride problem. *Expert Syst. Appl.* **2023**, *213*, 119044.
36. Zhang, D.; Zhou, Y.; Wang, J. Game theory-based approach for massive route planning in dynamic road networks. *IEEE Trans. Consum. Electron.* **2025**, *71*, 2131–2141.
37. Kuo, Y.; Wang, C.-C. A variable neighborhood search for the multi-depot vehicle routing problem with loading cost. *Expert Syst. Appl.* **2012**, *39*, 6949–6954.
38. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680.
39. Guo, X.; Zhou, M.; Liu, S.; Qi, L. Lexicographic multiobjective scatter search for the optimization of sequence-dependent selective disassembly subject to multiresource constraints. *IEEE Trans. Cybern.* **2019**, *50*, 3307–3317.
40. Laili, Y.; Li, Y.; Fang, Y.; Pham, D.T.; Zhang, L. Model review and algorithm comparison on multi-objective disassembly line balancing. *J. Manuf. Syst.* **2020**, *56*, 484–500.
41. Zhang, L.; Wu, Y.; Zhao, X.; Pan, S.; Li, Z.; Bao, H.; Tian, Y. A multi-objective two-sided disassembly line balancing optimization based on artificial bee colony algorithm: A case study of an automotive engine. *Int. J. Precis. Eng. Manuf. Green Technol.* **2022**, *9*, 1329–1347.
42. Ghannadpour, S.F.; Zarrabi, A. Multi-objective heterogeneous vehicle routing and scheduling problem with energy minimizing. *Swarm Evol. Comput.* **2019**, *44*, 728–747.
43. Zhao, P.; Luo, W.; Han, X. Time-dependent and bi-objective vehicle routing problem with time windows. *Adv. Prod. Eng. Manag.* **2019**, *14*, 201–212.
44. Wang, F.; Liao, F.; Li, Y.; Yan, X.; Chen, X. An ensemble learning based multi-objective evolutionary algorithm for the dynamic vehicle routing problem with time windows. *Comput. Ind. Eng.* **2021**, *154*, 107131.

45. Shao, W.; Shao, Z.; Pi, D. Multi-objective evolutionary algorithm based on multiple neighborhoods local search for multi-objective distributed hybrid flow shop scheduling problem. *Expert Syst. Appl.* **2021**, *183*, 115453.
46. Wang, L.; Peng, Z.-p. Solving energy-efficient distributed job shop scheduling via multi-objective evolutionary algorithm with decomposition. *Swarm Evol. Comput.* **2020**, *58*, 100745.
47. Pan, Y.; Gao, K.; Li, Z.; Wu, N. Solving biobjective distributed flow-shop scheduling problems with lot-streaming using an improved jaya algorithm. *IEEE Trans. Cybern.* **2022**, *53*, 3818–3828.
48. Lopez, P.A.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.-P.; Hilbrich, R.; Lücken, L.; Rummel, J.; Wagner, P.; Wießner, E. Microscopic traffic simulation using SUMO. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems, Maui, HI, USA, 4–7 November 2018; pp. 2575–2582.
49. Zhu, Y.; Tang, Q.; Zhang, L.; He, M.; Kapenda, J. Improved multi-objective artificial bee colony algorithm for parallel machine lot-streaming scheduling problem with limited and unequal sub-lots. *Comput. Ind. Eng.* **2023**, *183*, 109428.
50. Sun, Y.; Yen, G.G.; Yi, Z. IGD indicator-based evolutionary algorithm for many-objective optimization problems. *IEEE Trans. Evol. Comput.* **2018**, *23*, 173–187.
51. Shang, K.; Ishibuchi, H.; He, L.; Pang, L.M. A survey on the hypervolume indicator in evolutionary multiobjective optimization. *IEEE Trans. Evol. Comput.* **2020**, *25*, 1–20.
52. Ma, X.; Fu, Y.; Gao, K.; Zhang, H.; Mou, J. A knowledge-based multi-objective evolutionary algorithm for solving home health care routing and scheduling problems with multiple centers. *Appl. Soft Comput.* **2023**, *144*, 110491.
53. Montgomery, D.C. *Design and Analysis of Experiments*; John Wiley & Sons: Hoboken, NJ, USA, 2017.
54. Fu, Y.; Zhou, M.; Guo, X.; Qi, L. Scheduling dual-objective stochastic hybrid flow shop with deteriorating jobs via bi-population evolutionary algorithm. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *50*, 5037–5048.
55. Chang, P.C.; Chen, S.H.; Zhang, Q.; Lin, J.L. MOEA/D for flowshop scheduling problems. In Proceedings of the IEEE World Congress on Computational Intelligence, Hong Kong, China, 1–6 June 2008; pp. 1433–1438.
56. Fu, Y.; Zhang, Z.; Huang, M.; Guo, X.; Qi, L. Multi-objective integrated energy-efficient scheduling of distributed flexible job shop and vehicle routing by knowledge-and-learning-based hyper-heuristics. *IEEE Trans. Emerg. Top. Comput. Intell.* **2025**, *9*, 2137–2150.

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.