A Multiline Customized Bus Planning Method Based on Reinforcement Learning and Spatiotemporal Clustering Algorithm

Wengang Li, Linjiang Zheng, Longquan Liao, Xingze Yang, Dihua Sun, and Weining Liu

***Abstract*—The demand-responsive customized bus has been operated in real life, which is a crucial way to improve the service quality and efficiency of the urban public transportation system. Reasonable station and line planning can enhance customized bus competitiveness in residents' travel mode. Most previous studies on optimizing customized bus lines rely on historical passenger volume and travel time to generate static schemes, but the actual operation process of customized bus is often in uncertain circumstances, such as road congestion. The static strategy will occur deviations in this situation. This study proposes a novel planning method to address the above issue. First, a spatiotemporal clustering algorithm is proposed to generate joint stations based on the passenger travel demand. Second, the method models the multiline customized bus optimization problem as a Markov decision process and uses a multiagent deep reinforcement learning algorithm to ensure effective training and response to incomplete information scenarios. Finally, the rationality of the proposed planning method is verified in a case study of customized bus area in Chongqing, China. Compared with the latest heuristic optimization algorithm, our method can effectively reduce the operating and passenger costs in complex environments.**

***Index Terms*—Customized bus, deep reinforcement learning, line optimization, spatiotemporal clustering, station planning.**

I. Introduction

With the development of economy and society, the rapid expansion of cities and the increasing level of motorization have brought a series of traffic-related problems, such as congestion, accidents, pollution, and energy consumption. In response to this situation, an innovative demand-responsive public transportation service such as customized bus has been proposed. Due to the advantages of intensive efficiency, energy saving, and environmental protection, the customized bus has developed rapidly. The customized bus system designs the optimal travel time and route for passengers with similar travel time and travel origin-destination points. By providing convenient, affordable, and high-quality customized bus services, it can attract some passengers to give up transportation methods such as private car and convert to public transportation [1].

Station planning and line optimization play a crucial role in the process of customized bus services. Station planning is the basis of bus line design. Appropriate station distribution can effectively aggregate passengers' travel demands to achieve good results in bus line optimization. Customized bus line optimization is a process of mutual restriction and satisfaction between customized bus operators and passengers [2]. On the one hand, customized bus operators are guided by the travel needs of passengers while considering the interests of company and social benefits to design customized bus line. A reasonable planning scheme may prevent the operating company from meeting the demand for a single trip at a higher cost. On the other hand, passengers choose the appropriate or preferred mode of transportation from their perspectives based on the characteristics of existing travel modes. When the uncertainty of the line design affects the waiting time of passengers, whether passengers still choose customized bus service that will in turn affect the bus line optimization. Therefore, it is difficult to balance the service relationship between the operating company and individual demands in customized bus line optimization.

As a transportation mode still under development and exploration, the design of the customized bus line optimization has achieved some preliminary results. Existing conventional bus optimization methods, such as heuristic algorithms, have the advantages of simple design and efficient processing. But they belong to mechanical programming and lack self-learning ability. When dealing with some dynamic problems, these conventional methods are limited to adding dynamics during model construction, and they usually require a large number of assumptions to achieve reasonable results, which is difficult to verify in practical applications.

Reinforcement learning (RL) can be a potential approach to address the aforementioned challenges. Its basic principle is that the agent takes actions in the environment and adjusts strategies according to the feedback of the environment, to maximize total reward or complete specific goals. Previous research has

Manuscript received 15 June 2023; revised 18 September 2023; accepted 30 October 2023. Date of publication 30 November 2023; date of current version 31 May 2024. This work was supported in part by the National Key R&D Program of China under Grant 2022YFC3006400; in part by the Fundamental Research Funds for the Central Universities, China, under Grant 2022CDJDX-003; in part by the Transportation Science and Technology Program of Chongqing, China, under Grant 2022-09; and in part by the Science and Technology Innovation Key R&D Program of Chongqing, China, under Grant CSTB2022TIAD-KPX0099. *(Corresponding author: Linjiang Zheng.)*

Wengang Li, Linjiang Zheng, Longquan Liao, Xingze Yang, and Weining Liu are with the College of Computer Science, Chongqing University, Chongqing 400044, China, and also with the Key Laboratory of Dependable Service Computing in Cyber Physical Society of the Ministry of Education, Chongqing University, Chongqing 400044, China (e-mail: zlj_cqu@cqu.edu.cn).

Dihua Sun is with the Department of Automation, Chongqing University, Chongqing 400044, China.

Digital Object Identifier 10.1109/TCSS.2023.3329990

successfully applied RL to practical problems, such as multirobot autonomous exploration [3] and UAV trajectory planning [4], [5]. The multiagent RL (MARL) framework can allow multiple agents to share experience and cooperate to complete complex tasks at a low cost and high efficiency, which makes MARL more suitable for collaborative or competitive problems. In MARL, all agents simultaneously take their own actions based on the current state in the environment. The joint actions formed by their respective actions affect the update of the environment and determine the reward received by each agent. This mechanism is highly similar to the optimization problem of multiple bus lines interweaving to serve passengers. Inspired by the successful application of multiagent RL framework to the multirobot exploration problem, we attempt to solve the multiline customized bus problem from a data-driven perspective.

The key contributions of this work are as follows.

1) We introduce the characteristics of customized bus and establish a customized bus planning model from the perspective of passengers' travel time and operating costs.
2) According to the travel time distribution of passengers and the distribution of travel starting and ending locations, we propose a joint station planning method based on the spatiotemporal clustering algorithm.
3) We model the multiline customized bus optimization problem as a Markov decision process and utilize a multiagent deep reinforcement learning method, which includes the design of action, state, reward function, and other components, to ensure effective training and response to uncertain road circumstances.
4) We obtain a customized bus line planning scheme based on the actual passenger travel demand and road network, which verifies the effectiveness of the proposed method in a case study. Compared with the widely used heuristic algorithm, the RL method can effectively reduce travel costs and improve travel efficiency. In the final section of the case study, we designed a dynamic environment to demonstrate that the proposed method can perform well even in uncertain scenarios.

II. Literature Review

The customized bus and ride-sharing are considered as the type of Mobility as a Service (MaaS). The ride-sharing is a peer-to-peer transport service, and passengers only receive service at the prearranged location. It focuses on how to transport all passengers with minimal cost and is usually suitable for small car-sized groups that the number of passenger is often less than five. The customized bus is a door-to-door travel mode and aims at large-scale travel groups that the number of passenger is more than 30. Therefore, passengers must receive service at a designated common location (joint station). The customized bus problem not only focuses on how to pick up and drop off passengers with the minimum cost but also focuses on how to set reasonable stations for pick up and drop off passengers. As such, this study mainly involves the two branches of existing literature.

A. Customized Bus Station Planning Method

Unlike the conventional bus, the customized bus needs to obtain the passengers' travel demands in advance and plan appropriate stations for passengers, which ensures that the customized system provides reliable services to passengers.

In most cases, station planning methods are applicable to static travel demand [6]. These methods require that passengers must arrive at the bus station to receive service at the designated time. This assumption simplifies the problem. Han et al. [7] collected historical individual travel booking demand data and presented a detailed flow chart of a CB network planning methodology. Qiu et al. [8] extracted the potential CCB passengers from regular bus passengers based on the bus smart card data and proposed a pairwise density-based spatial clustering algorithm to identify the origin and destination distribution of potential CCB passengers. Meanwhile, the discovered hot locations of potential CCB passengers can be regarded as the candidate locations of CCB stations and can be used to set candidate CCB lines. He and Song [9] proposed a microevolutionary algorithm for the dynamic location problem of customized bus stops with maximum distance constraints.

Some scholars have studied the customized bus [10], [11], [12] by collecting the passengers' dynamic travel demand and integrating the method of station planning into the dynamic interaction between vehicles and passengers. Shu and Li [13] proposed an improved K-means clustering algorithm to plan bus stations for responsive dynamic demand. The advantage of this clustering algorithm is that the long-distance cart sites can be located in the most crowded positions of passengers, and the distance system between passengers and stations is optimal, which is convenient for passengers to arrive at the stations.

In summary, common station planning methods include agglomerative hierarchical clustering (AHC), DBSCAN, and K-means. These methods only consider the spatial factor for station planning, while ignoring the temporal factor. Most studies directly consider temporal factors in line optimization. However, the existence of temporal outliers inevitably leads operating companies to dispatch multiple vehicles to meet the decentralized demands, which increases operating costs and overall passengers' travel costs in line optimization. Therefore, we propose a spatiotemporal clustering algorithm for station planning to address the aforementioned issue. As we known, this article is the first study to consider removing temporal outliers in the customized bus problem, and we demonstrated the effectiveness and necessity of this method through case study.

B. Customized Bus Line Optimization Model

The line optimization problem of the customized bus is a special vehicle line optimization problem that is nondeterministic polynomial-time hard (NP-hard) [14].

When analyzing cases with low travel demand, exact algorithms are usually used, such as dynamic programming, network flow algorithm, branch and bound algorithm, greedy algorithm, and column method [15], [16], [17]. Lyu et al. [18] proposed a CB line planning framework called CB-Planner, which combines grid density with the heuristic algorithm to generate CB

TABLE I
EXISTING WORKS ON CUSTOMIZED BUS

Research Objectives	Research Types	Algorithm	Literature
Station planning	Static demand	DBSCAN	[6], [8]
		Agglomerative Hierarchical Clustering	[7]
		Microevolution algorithm(MEA)	[9]
	Dynamic demand	K-means	[13], [18], [20]
Line optimization	Exact algorithms	Branch-and-bound (B&B) algorithm	[10], [16]
		Dynamic programming	[12], [15], [17], [18]
	Heuristic algorithms	Genetic Algorithm (GA)	[6], [24], [25], [33], [34]
		Nondominated Sorting Genetic Algorithm II (NSGA-II)	[11], [26], [34]
		Tabu search	[19], [21]
		Ant colony optimization (ACO)	[7], [20], [22]
		Particle swarm optimization (PSO)	[23]
	Reinforcement learning	Q-learning	[32]

lines. Considering that these exact algorithms usually require strict constraints and the computing time increases rapidly with the increase of the data size, practical problems for large-scale demand can only be solved by heuristic algorithms. Genetic algorithms are the most commonly used optimization methods. Other heuristic algorithms include ant colony algorithm, particle swarm algorithm, and tabu search algorithm [19], [20], [21], [22], [23]. Chen et al. [24] developed an integrated optimization method for CB stop deployment, route design, and timetable development optimization problems while meeting travel demands as much as possible to obtain system-optimal CB service plans. A heuristic approach is applied to generate efficient solutions for the CB service design problem. Wang et al. [25] proposed the vehicle scheduling methods of multiple customized bus routes with passenger travel time window constraints and established a customized bus scheduling optimization model with a genetic algorithm to solve, but did not involve the robust model. Ma et al. [26] proposed a three-stage hybrid coding method based on the NSGA-II algorithm to deal with customized bus route optimization under uncertain conditions. With the Bertsimas-Sim robust optimization theory, the robust peer-to-peer transformation was performed on the robust model with uncertain parameters.

In addition, some scholars have carried out research on line network optimization based on nonparametric models [27], [28]. Peng et al. [29] proposed a multiple route planning model (multiroute dynamic programming model) to solve the urban route planning problem with traffic flow information. The reward function suitable for urban path planning is designed, and multiple routes are generated based on the Q values. Due to the intelligent nature of RL methods, the DP model has great potential to solve the problem with more complicated route planning objectives. Wang et al. [30] designed a restricted reinforcement learning-artificial potential field (RRL-APF) algorithm to establish a risk aversion path planning model. Geng et al. [31] designed a route planning algorithm based on deep reinforcement learning (DRL) for pedestrians.

Due to the Q-learning algorithm has the problems of slow solving speed and low efficiency, traditional reinforcement learning is rarely combined in the research of customized bus line optimization [32]. As the improvement of solution efficiency of deep reinforcement learning, this article proposes a method for configuring and optimizing multi route customized public transportation plans using a multiagent DRL framework. Owning to its advantage of the low requirements for environmental models and strong self-updating ability, the method performed better in our case study. The proposed RL framework will be a foundation of the future studies.

Table I clearly shows the relevant research methods about station planning and line optimization on the customized bus problem.

III. CUSTOMIZED BUS OPTIMIZATION MODEL

A. Problem Description

Most of the customized bus groups in the morning rush hours are commuters, whose travel modes are relatively stable in time and space dimensions and tend to travel from residential areas to working areas. Therefore, we first need to find passengers with spatiotemporal correlation and plan appropriate stations for them. Assuming a predefined maximum distance that the passengers are willing to walk, the stations within the walking area are considered candidate boarding and alighting places. A suitable coverage is divided for each candidate boarding and alighting station, and the demand points within the scope are potential customized bus groups.

The scene diagram of customized bus is shown in Fig. 1. Due to the fact that customized bus is a door-to-door transportation service [35], the boarding and alighting stations belong to different non overlapping areas. Passengers can only board at the boarding area and alight at the alighting area. Vehicles pass quickly midway and passengers are not allowed to board or alight midway. The operation lines in the boarding area and alighting area are solved respectively. When solving the line of the boarding area, $W = \{w_1, w_2, \ldots, w_i\}$ is defined as the

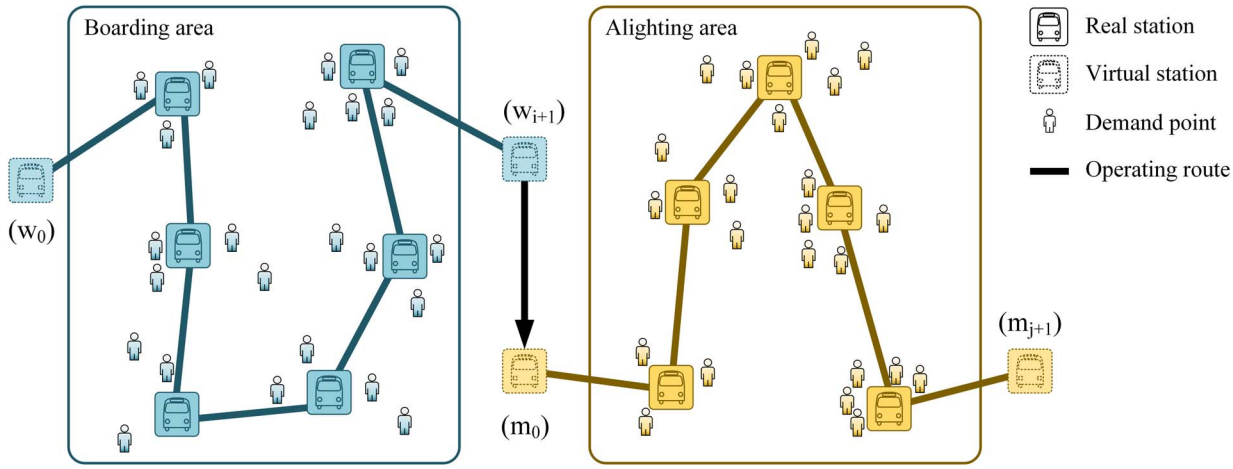


Fig. 1. Scene diagram of customized bus.

TABLE II
SYMBOL DEFINITION

Symbol	Type	Definition
x_{ij}^k	Variable	When vehicle k running from station i to station j, x_{ij}^k is 1; otherwise 0
y_s^k	Variable	When vehicle k stops at station s, y_s^k is 1; otherwise 0
b_s^k	Variable	Number of passengers boarding when vehicle k stops at station s
u_s^k	Variable	Number of passengers alighting when vehicle k stops at station s
t_{pw}	Parameter	Waiting time of passenger at the station
t_{pl}	Parameter	The time later than the arrival time window of passengers who arrive at their destination
T_{ij}	Parameter	Vehicle travel time from departure station i to destination station j
l_{ij}	Variable	Vehicle running distance from departure station i to destination station j
C_k	Parameter	Passenger capacity of vehicle k
$l_{\min}$	Parameter	Minimum running distance of the line
$l_{\max}$	Parameter	Maximum running distance of the line
S	Set	Station collection, including boarding stations and alighting stations
W	Set	Collection of boarding stations
M	Set	Collection of alighting stations
K	Set	Collection of customized bus

collection of real stations in the boarding area, and the virtual station w_0 is set as the starting station of the line. Define the virtual station w_{i+1} as the terminal of the line. The departure time of the alighting area is the time when the vehicle arrives at the terminal of the boarding area. When solving the line of the alighting area, $M = \{m_1, m_2, \ldots, m_j\}$ is defined as the collection of real stations in the alighting area, and the virtual station m_0 is considered the starting station of the alighting area. The virtual stop m_{j+1} is defined as the terminal station of the alighting area.

B. Model Construction

Table II lists the mathematical symbols and notations used in this article.

To facilitate modeling and solution, the following conditions are assumed to be true in the model:

1) Each travel demand only corresponds to one boarding station and alighting station;
2) During the customized bus operation, each vehicle can stop at multiple stations, but each station can only stop once. Vehicles are not allowed to pass through the already stopped stations again;
3) The customized bus runs in the road network at a constant speed. Assuming that the distance between stations is known and fixed, the running time of vehicles between stations is determined.

How to determine whether passengers accept the customized bus service? Inspired by [13], we introduce a time window to determine whether there is a service relationship between the operating company or passengers. There are two types of time window.

1) *Hard time window*: Passengers are very sensitive to time and can only accept bus services within the time window. Once the arrival time of the customized bus is not within the time window, passengers will no longer accept bus services.
2) *Soft time window*: Passengers are not sensitive to time and are willing to accept services outside the time window. The time cost should be calculated in this case.

We use a hard time window to measure whether passengers accept customized bus services during the boarding stage, and define the passenger's waiting time window as WTW. We use a soft time window to measure the time cost of passengers during the alighting stage. Passengers should arrive at the station earlier than the vehicle and accept a certain WTW. If the vehicle arrives later than WTW, passengers will give up the service. If the passenger chooses a customized bus service, it is considered that the service cannot be canceled. When the vehicle arrives at the destination later than the passenger's arrival time window, the passenger's time cost is calculated in the model.

1) Objective Function: As with the conventional bus, the line design of customized bus also needs to determine the line layout and relevant attributes by optimizing specific objective functions. From the passenger's perspective, the objectives of customized bus line optimization are shorter travel time [36], larger service scope, better accessibility, and reliability. However,

from the operator's perspective, the goal of customized bus line optimization is to attract as many passengers as possible while reasonably controlling the line length and reducing operating costs. Therefore, it is necessary to pay attention to and balance the interests of both passengers and operating companies to obtain a more reasonable line planning scheme to minimize the operating cost in customized bus problem.

Passenger travel cost: The passengers' travel time mainly consists of the following two aspects: passengers' waiting time and passengers' late arrival time. Based on our assumption, passengers will arrive at the station early and wait at WTW. To avoid the vehicle arriving later than the passenger's arrival time window, the customized bus needs to deliver passengers in the shortest possible time, which means all passengers need to spend the least time when the vehicle is driving. The passengers' travel time cost can be expressed as z_1

$$z_1 = \alpha_1 \left(\sum_{k\in K} \sum_{i\in W} y_i^k b_i^k t_{pw} \right) + \alpha_2 \left(\sum_{k\in K} \sum_{j\in M} y_j^k u_j^k t_{pl} \right). \tag{1}$$

Vehicle operating cost: The variable operating cost of the vehicle is positively related to the running time of the vehicle, including the running time of the vehicle between stations and the time spent by the vehicle to get on or get off passengers. The vehicle operating cost can be expressed as z_2

$$z_2 = \beta \left(\sum_{k\in K} \sum_{i\in S} \sum_{j\in S} (x_{ij}^k T_{ij} + t_s) \right) \tag{2}$$

$$t_s = \max \left\{ \delta_1 \sum_{i\in W} y_i^k b_i^k, \delta_2 \sum_{j\in M} y_j^k u_j^k \right\}. \tag{3}$$

Note that z_1 and z_2 are related. When the vehicle operating cost increases, the late arrival time cost may also increase. If vehicles spend more time traveling, passengers are more likely to arrive late at their destination. But it is not absolute, the vehicle may sacrifice the arrival time cost of some passengers and deliver other passengers on time by planning a specific route in order to achieve the optimal total time cost. If only the vehicle operating cost is considered, it may fall into a local optimal solution. Therefore, we introduce the late arrival time cost in the objective function z_1 and adjust the parameter a_2 to avoid falling into a local optimal solution.

There is no single solution to simultaneously optimize each objective for the multiobjective optimization problem composed of the time costs of passengers and vehicles. Therefore, we transform multiobjective optimization into single objective optimization and establish a customized bus optimization model

$$\min \quad Z = z_1 + z_2. \tag{4}$$

2) Constraints: Accordingly, the model should meet the following constraints.

The length of line: Generally speaking, the line length of customized bus should be within a certain range. If the length of the line is too long, there will be an increase in uncertainty and a lack of punctuality of vehicles

$$l_{\min} \le \sum_{i\in S} \sum_{j\in S} l_{ij} \le l_{\max}. \tag{5}$$

The capacity of vehicle: Compared with the conventional bus, the customized bus can ensure that all passengers on the bus have seats. Therefore, it is necessary to ensure that the actual carrying capacity of vehicles is not less than the number of passengers. The specific formula is as follows:

$$\sum_{k\in K} \sum_{i\in W} b_i^k \le C_k. \tag{6}$$

The number of stations: The number of stations for customized bus should not be too many, or it will not meet the passengers' travel timeliness. Therefore, the number of stations for customized bus needs to be limited

$$\sum_{s\in S} y_s^k \le \varphi \qquad \forall k \in K. \tag{7}$$

IV. Solving Algorithm

A. Clustering for Customized Bus Stations

Before establishing the line optimization model, the location of the boarding and alighting stations in the service area should be generated. The actual passenger demand position will be randomly distributed in the whole service area, and the travel time of passengers is not centralized, which leads to the unsatisfactory effect of the traditional spatial clustering algorithm. The existence of spatiotemporal outliers may cause the customized bus to satisfy a single demand at a high cost, which seriously affects the efficiency of the customized bus.

We consider the limitations of the current research and propose the spatiotemporal clustering algorithm, named the spatiotemporal clustering algorithm based on DBSCAN and PAM (ST-DP), which is implemented using the principles of the DBSCAN algorithm and PAM algorithm. The DBSCAN clustering algorithm divides travel groups according to the degree of association of travel demand points. The basic idea is: the number of objects contained in a certain area of the clustering space is not less than a given threshold *MinPts*, and each object in the same area is reachable. The principle of the PAM algorithm is to divide a given sample set into K clusters according to the distance between samples. Make the points in the cluster as close as possible, and make the distance between cluster groups as large as possible.

The flowchart of the ST-DP algorithm is shown in Fig. 2, and its basic steps are as follows:

Step 1: Set temporal distance threshold *temporal_threshold*, spatial distance threshold *spatial_threshold*, and the number of spatiotemporal object threshold *MinPts*. The first two parameters are used to determine the spatiotemporal neighborhood domain, and the last one is used to determine the number of objects in the spatiotemporal neighborhood domain.

Step 2: Input the original passenger demand, which includes the spatiotemporal information of each passenger.

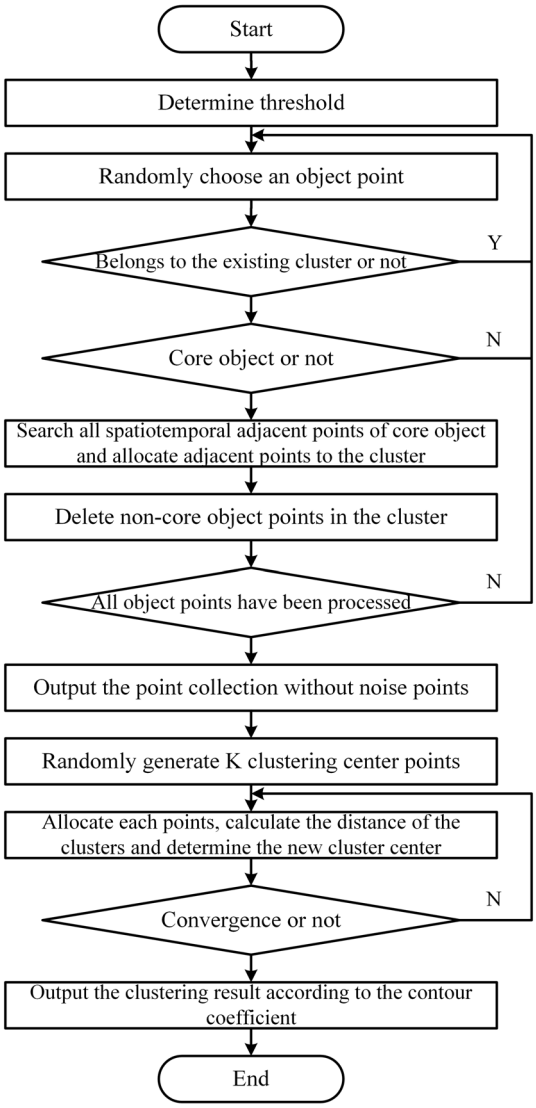


Fig. 2. ST-DP clustering algorithm flowchart.

The information is recorded as an object point $P_i = \{x_i, y_i, t_i\}$, which indicates that the ith passenger is at time t_i starts from the coordinate (x_i, y_i). The collection of all object points is D_p.

For two spatiotemporal object points $P_1 = \{x_1, y_1, t_1\}$ and $P_2 = \{x_2, y_2, t_2\}$, their temporal distance is

$$\nabla T = |t_1 - t_2| \tag{8}$$

and spatial distance is

$$\nabla S = \sqrt{(x_1 - x_2)^2 - (y_1 - y_2)^2}. \tag{9}$$

When two spatiotemporal object points satisfy $\Delta T <$ *temporal_threshold* and $\Delta S <$ *spatial_threshold*, they are considered directly density reachable points.

Step 3: Select one object point P_i in sequence from D_p. Determine whether it belongs to the existing cluster. If yes, select the next object point again. Otherwise, go to Step 4.

Step 4: Find all objects in the spatiotemporal neighborhood of P_i. If the number of the objects more than *MinPts*, P_i is the spatiotemporal core point, then go to Step 5; otherwise, go back to Step 3 and select the next object point.

Step 5: Search all adjacent points Q_i of spatiotemporal core object points. If it does not belong to any existing cluster group, set it into the new cluster group. If it belongs to an existing cluster group, no operation is performed.

Step 6: Determine whether the newly added object in cluster C is a spatiotemporal core object. If it is not a spatiotemporal core object, mark it as an edge spatiotemporal object without further operation. If it is, repeat Step 5 for the spatiotemporal core object.

Step 7: Repeat Steps 3–6 until all objects in D_p belong to a cluster or are spatiotemporal outliers.

Step 8: According to the calculation results in Step 7, completely eliminate the data identified as noise points, and use the remaining demand point data as a new demand dataset. The coordinates of each point A_i are (x_i^A, y_i^A), and the coordinates of the cluster center point C_j are (x_j^C, y_j^C).

Step 9: Set the value K of the clusters algorithm and randomly select K point as the initial cluster center points C_j.

Step 10: Calculate the distance from the sample data to the cluster center C_j

$$\mathrm{Dis}(A_i, C_j) = \sqrt{(x_i^A - x_j^C)^2 - (y_i^A - y_j^C)^2}. \tag{10}$$

Then cluster each point to the nearest cluster center

$$\mathrm{Dis}(A_i, C_k) = \mathrm{minDis}(A_i, C_j). \tag{11}$$

Step 11: For each cluster, calculate the distance E from all points except the cluster center to other points, and take the point with the smallest E as the new cluster center C'_j.

Step 12: Calculate the sum of squares of errors

$$\mathrm{SSE} = \sum \sum \mathrm{Dis}(A_i, C_j). \tag{12}$$

Step 13: Calculate iteratively SSE. When the variation range of SSE is less than the accuracy, the iteration ends.

Step 14: Calculate the contour coefficient of the current clustering result, adjust the K value, and repeat Steps 9–13 until the optimal clustering result under the station's constraint is obtained.

The advantage of the proposed clustering algorithm in this study is that the algorithm automatically generates different travel clusters according to the density relationship of demand points. In this problem, the *temporal_threshold* represents the similarity of departure time or arrival time of customized bus passengers; the *spatial_threshold* represents the proximity of the residential or working places of the customized bus passenger cluster; the number of spatiotemporal object threshold *MinPts* reflects the minimum travel demand carrying value of customized bus stations, which can accurately describe the travel demand of passengers in reality. In addition, the spatiotemporal clustering algorithm has better robustness to noise and isolated points [37].

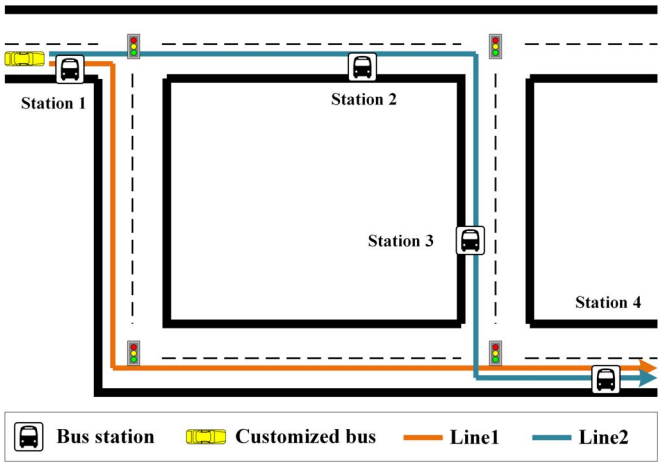


Fig. 3. Customized bus operation in the road network.

B. Search Strategy of the Algorithm

1) RL Framework: To obtain the optimal customized bus lines, we first define the basic elements of reinforcement learning as follows:

Agent: In the field of MDP, the agent is a virtual entity that senses the environment, takes action and maintains the value function in reinforcement learning, which is represented as vehicle k in customized bus. In each step t, the agent uses the current state $s_t \in S$ as the input of the model and then uses the strategy to select action $a_t \in A$, where policy is a mapping from the state space to the action space [38]. The agent executes action a_t, a reward $r_t \in R$ is received, and the state transits to the next state s_{t+1}, which can be represented by $p(s', r \mid s, a) = \Pr(s_{t+1} = s', r_t = r \mid s_t = s, a_t = a)$.

The multiagent system allows multiple individuals with simple intelligence to cooperate to explore an unknown environment to achieve goals while each agent constantly learns and updates its strategy. In this example, the coupled bus stations are selected for simultaneous optimization. Each bus should have a unique strategy to meet the requirements because each bus lines have different stop time, stop station, and passenger capacity. Therefore, we use a multiagent framework to represent each bus with an agent. The number of agents is equal to the number of buses, which is 3 in this study.

Action design: In this study, agents are used to determine the sequence of customized bus stations. In previous study [32], different actions indicated that the vehicle is going to different stations. However, the actual situation is shown in Fig. 3.

Although the agent can directly arrive at station 4 from station 1 along line 1, the learning experience shows that there are more possibilities for vehicles to drive along line 2 to obtain better comprehensive benefits with less time cost. We design the action space as different stations and whether or not to stop at the station to distinguish the vehicle is passing or stopping at a certain station. The action can be represented as $a = (next\ station, y_s^k)$. The status y_s^k can be expressed as 1 when the vehicle stops at a certain station, and boarding or alighting passengers. In this case, the agent takes action $< (s_2, 0), (s_3, 0), (s_4, 1)>$ may receive more total rewards than takes action $<(s_4, 1)>$. The condition for the end of an episode in the boarding area is $next\ station = w_{i+1}$, and the condition for the end of a completed episode is $next\ station = m_{i+1}$.

State design: In the customized bus problem, the state is usually considered the information of bus stations determined by clustering algorithm or other methods. If only the stations are used to represent the state, then the joint process of the state and the action does not conform to the Markov property. Therefore, we introduce other information into the state design. The state can be represented as $s = (current\ station, the\ number\ of\ onboard\ passengers, vehicle\ travel\ time)$, and $s_0 = (w_0, 0, 0)$.

Reward function design: the reward is used to evaluate the performance of each action of the agent. The design of reward function will affect the efficiency and effectiveness of the learning process. In the previous section, we defined the objective function in the model, which includes vehicle operation and passenger time cost. The reward function also considers these two modules. For each action step t, the reward function r_t can be expressed as

$$r_{t1} = -\alpha' \left(\alpha_1 \sum_{i \in W} B(i,t) + \alpha_2 \sum_{j \in M} U(j,t) \right) \tag{13}$$

$$r_{t2} = -\beta' \sum_{k \in K} T(k,t) \tag{14}$$

$$r_t = r_{t1} + r_{t2} \tag{15}$$

where $B(i, t)$ is the waiting time of passengers who wait at boarding station w_i on all lines at step t. $U(j, t)$ is the time later than the arrival time window of passengers who get off at station m_j of all lines at step t. $T(k, t)$ is the travel time of the kth vehicle at step t. The relationship between the reward function and objective function can be expressed as

$$z_1 = -\frac{1}{\alpha'} \sum_t r_{t1} \tag{16}$$

$$z_2 = -\frac{1}{\beta'} \sum_t r_{t2}\,. \tag{17}$$

Generally, the discounted cumulative reward can be formulated as the expected return R_t as

$$R_t = \sum_{i=0} \gamma^i r_{t+i} \tag{18}$$

where γ $(0 < \gamma \leq 1)$ represents the discount factor.

Simulator: we design a simulator that explicitly describes how the basic elements of reinforcement learning process in multiagent systems. As shown in Algorithm 1, each step includes the following works: agent observation, agent decision, action execution, environment feedback, state transition, and learning.

2) RL Algorithm: The design of action and state space in this model will lead to high costs for the classical reinforcement

Algorithm 1 Simulator for a multiagent system

Input: travel demand information for all passengers
Initialize joint states s_0.
for $t = 1$ to T **do**:
  Agent Observation: Each agent observes the environment and obtains information which includes the number of passengers waiting at all stations, the waiting time for each passenger within step t, and so on.
  Agent Decision: Based on observed environmental information, each agent independently decides what action a_t to take.
  Action Execution: Each agent takes the action a_t and applies it to the environment.
  Environment Feedback: After each agent executes the action a_t, the environment generates feedback, which is a reward. For the step t, the reward is $r_t = r_{t1} + r_{t2}$. The definitions of r_{t1} and r_{t2} are detailed in (13) and (14), respectively.
  State Transition: Each agent transitions to the next state s_{t+1}.
  Learning and Policy Update: Based on a centralized learning framework, each agent updates its strategy. The specific update strategy is shown in Algorithm 2.
end for

Algorithm 2 MADQN

Initialize replay memory D
Use random weight θ to initialize the network
while the number of epochs **do**:
  Reset the environment and obtain the initial joint state s_0
  for $t = 1$ to T **do**
    for $k = 1$ **to** K **do**
      Sample action of each agent a_t^k, according to greedy policy
    end for
    According to the action a_t, then observe reward r_t and next state s_{t+1}
    Store transitions of all agents $(s_t^k, a_t^k, r_t^k, s_{t+1}^k)$, $\forall\, k = 1, \ldots, K$ in the replay memory D
  end for
  for $k = 1$ **to** K **do**
    Sample a mini-batch of transitions from replay memory D
    Calculate target value according to (22)
    Update the parameters of Q-network by: $\theta \leftarrow \theta + \beta_1 \nabla_\theta [L(\theta)]^2$
  end for
end while

learning algorithm. Therefore, a more effective off-policy algorithm is used to deal with the RL problem of discrete action space.

Q function: In RL, the state-action value function $Q(s, a)$ refers to the execution of action a in the current state s and always follows the policy π until the end of the episode, and the cumulative returns obtained by the agent in this process are denoted as follows:

$$Q^*(s,a) = \max_{\pi} E[R_t | s_t = s, a_t = a, \pi]. \tag{19}$$

The optimal Q function follows an important rule called Bellman equation, that is:

$$Q^*(s_t, a_t) = E_{s_{t+1}}[r_t + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})]. \tag{20}$$

It contains two parts: 1) the reward of the state and 2) the optimal action value $Q^*(s_{t+1}, a_{t+1})$ at the next step multiplied by the discount factor.

MADQN: The goal of reinforcement learning is to find the optimal strategy for one or more agents, which can select the next action by maximizing the expected value of the $r+\gamma Q^*(s_{t+1}, a_{t+1})$. The Q-learning is a widely used method for single agent applications. A wide extension of Q-learning is Deep Q-network (DQN). In the DQN method, the state and action are regarded as the inputs of the neural network, and the neural network is trained to fit the optimal action-value function (Q function).

In the MADQN architecture, a centralized learning framework is applied to multiagent system, that is, a neural network is used to model the behavior of all agents, which means the weights of the centralized neural network are shared by all agents [39]. The parameter of centralized network θ is updated by minimizing the total difference between the predicted value of the Q-network and the target Q value, as shown in

$$L(\theta) = E\left[\left(Q(s_t^k, a_t^k \mid \theta) - y\right)^2\right]. \tag{21}$$

If the current state is the termination state, the target Q value is estimated by the obtained reward. Otherwise, the target Q value is estimated by the sum of the obtained reward and the discounted estimated value of the next state, as shown in

$$y = \begin{cases} r_t^k, & \text{if} \quad \text{is\_end} = 1 \\ r_t^k + \gamma \max\limits_{a_{t+1}^k} Q(s_{t+1}^k, a_{t+1}^k), & \text{if} \quad \text{is\_end} = 0. \end{cases} \tag{22}$$

The details of the MADQN algorithm used in this article are shown in Algorithm 2. The model introduced a technique called *Replay Memory* to make the network updates more stable. In each step t, we first sample the actions of each agent with the greedy strategy based on the centralized Q network, then return the rewards and the next state of all agents after executing the actions, and store the transitions (s_t, a_t, r_t, s_{t+1}) of each agent in replay memory.

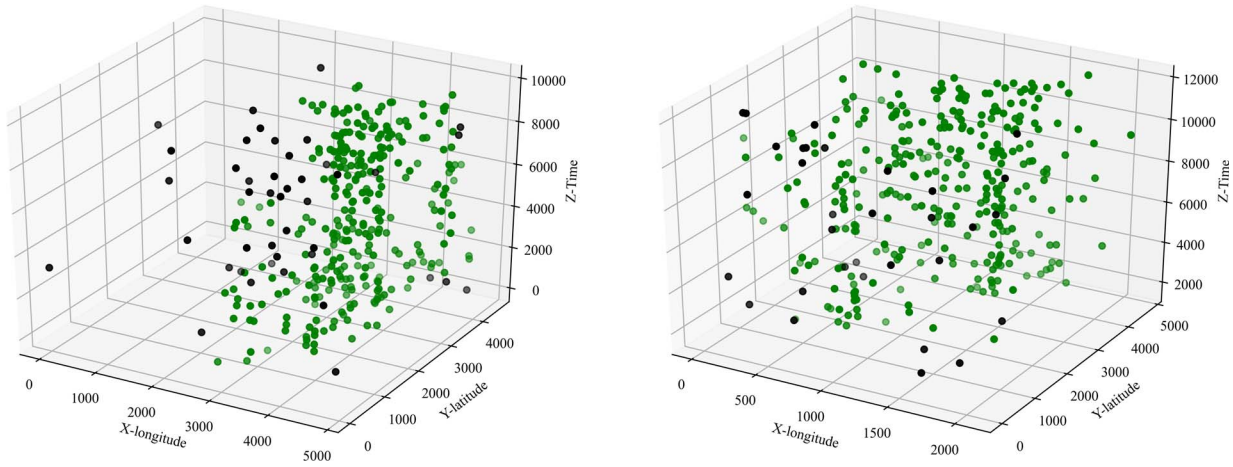


Fig. 4. Distribution of passengers' travel points (the left is starting point and the right is ending point). The black points are spatiotemporal outliers in ST-DP algorithm, representing noncustomized passengers. The green points represent the customized bus passengers.

TABLE III
TIME DISTRIBUTION OF PASSENGER "001"

Days	Departure Time	Arrival Time
Monday	07: 10: 25	07: 54: 44
Tuesday	07: 11: 49	07: 54: 56
Wednesday	07: 03: 47	07: 48: 09
Thursday	07: 10: 26	07: 52: 47
Friday	07: 00: 43	07: 42: 55
Representative time	07: 10: 25	07: 53: 45

To train the centralized Q-network (Steps 13–15 of Algorithm 2), we randomly sample a mini-batch of transitions from the replay memory and minimize the loss function in (21) at each episode to update the network parameter θ.

V. CASE STUDY

A. Customized Bus Stations Settings

We use the commuter travel demand data to carry out the case study. Based on the study of residential and working areas in the urban area of Chongqing, China, the study area is divided into Nan'an District and Yuzhong District. Nan'an District is the most typical residential distribution area in Chongqing, which is set as the boarding area. Yuzhong District is a typical working distribution area, which is set as the alighting area.

The passenger data information includes the latitude and longitude of the starting point, the latitude and longitude of the ending point, and the departure and arrival time. Table III shows the time distribution of passenger "001" in weekdays. The average value of the dates with the smallest difference in departure time is defined as the passenger's representative time, as the input of the clustering method. The following analysis is also based on the value of the passenger's representative time.

1) Cluster for Bus Stations: First, we conduct a time-space analysis of the departure and arrival locations of the commuter data in the morning peak. The departure time of passengers is distributed from 7:00 to 9:30, and the arrival time is distributed from 7:30 to 10:00. The starting points are concentrated, and the ending points are discrete. We use the coordinate conversion function to convert the longitude and latitude of the passenger's travel location into the planar coordinates and calculate the Euclidean distance between passenger travel demand points. Then, the ST-DP algorithm is used to eliminate the noise points with large discrete degree in the dataset to avoid the impact of noise points on the station settings. Based on the experience of conventional bus coverage division, the spatial threshold is set to 500 m, the temporal threshold is set to 10 min and the MinPts is set to 5. As shown in Fig. 4, we find that the noise points in clustering results have obvious outlier characteristics.

The silhouette coefficient and the predefined maximum distance are used to determine the appropriate parameter K in the ST-DP algorithm. The denser the demand points in the same cluster group and the distance between different cluster groups is far enough, the better the clustering effect is, and the larger the silhouette coefficient value is. The predefined maximum distance ensures that the maximum walking distance from the nearest station for all passengers is acceptable, which means that the distance from each cluster center point to each point in the cluster must be less than this value. Inspired by [8], the predefined maximum distance is set to 1 km. Comprehensively consider the station constraints and compare the silhouette coefficient under different K values. When K is 8, the starting travel point can achieve a better clustering result. When K is 9, the ending travel point can achieve a better clustering result. The clustering diagram is shown in Fig. 5.

2) Cluster Result Analysis: We determine the distribution of customized bus stations based on the clustering results and the actual situation. It includes eight boarding stations (marked as No.1 to No.8) and nine alighting stations (marked as No.9 to No.17). As shown in Table IV, the table contains four parts: the first column is the serial number of the stations in the boarding area; the second column is the number of passengers loaded at the boarding stations; the third column is the serial number of the stations in the alighting area; and the fourth column is the number of passengers unloaded at the alighting stations.

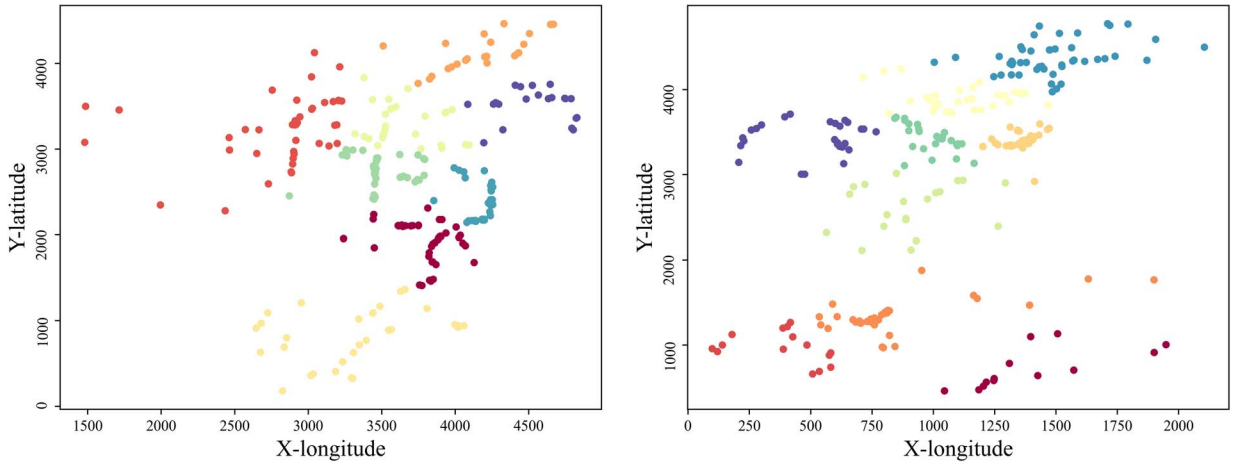


Fig. 5. ST-DP clustering result of passengers' travel points (the left is the starting point and the right is the ending point).

TABLE IV
STATION DISTRIBUTION OF CUSTOMIZED BUS

Boarding Station Serial Number	Number of Passengers Boarding	Alighting Station Serial Number	Number of Passengers Alighting
1	80	9	27
2	67	10	30
3	32	11	36
4	47	12	72
5	71	13	82
6	62	14	18
7	79	15	97
8	41	16	75
		17	42

We count the number of passengers per 10 min, as shown in Fig. 6. The number of commuters is basically stable but fluctuates at some times, passengers' working hours are not fixed. The blue line shows the passengers' mean travel time at each time, which also reflects the actual travel situation. The mean travel time spent by passengers who depart from 7:00 to 8:00 gradually increases, and the passengers who depart at 8:00 spend the most time. This is because the traffic network will be congested since 8:00, which will increase the travel time, and this situation will ease with the increase of time. Therefore, in the process of the model, we need to consider the speed changes caused by congestion.

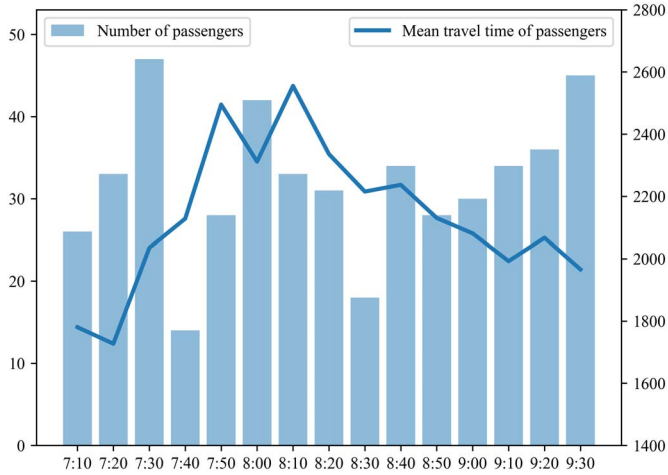


Fig. 6. Distribution of original passengers' trip over time.

TABLE V
DETAILS OF EACH TRAVEL GROUP

Group Serial Number	Departure Time	Number of Passengers	Vehicle Running Speed
Group 1	7:00–7:30	106	31 km/h
Group 2	7:30–8:00	84	25 km/h
Group 3	8:00–8:30	82	20 km/h
Group 4	8:30–9:00	92	27 km/h
Group 5	9:00–9:30	115	23 km/h

B. Customized Bus Scheme

The time span of commuters ranges from 1 October 2021 to 30 October 2021. In fact, commuters do not choose the same way to travel every day, which leads to the travel chain of the same commuter will not same every day. We divide the obtained data into training and testing datasets to verify the performance of the model. The training dataset consists of the working day data of the first three weeks. The passenger demands are randomly chosen from the training set for each training episode. The remaining data of the fourth week are used as the testing dataset.

The number of drivers is often limited in real life, we cannot assume the operator provide unlimited vehicles to meet the corresponding demand, which will make the model more uncertain in the application of actual scenarios. Considering the obtained passenger demand and passenger time cost, the line planning scheme is provided with 30 min as the timetable. We divide the passenger dataset into five subdatasets based on their departure time and arrange a certain number of vehicles for passengers in each timetable to provide customized services. The specific details are shown in Table V. In order to meet (6), the number of

TABLE VI
THE RESULT OF CUSTOMIZED BUS SCHEME SET AT WTW = 15 min

Group Serial Number	Line1				Line2				Line3			
	Via Station	AR (%)	WT (s)	RT (s)	Via Station	AR (%)	WT (s)	RT (s)	Via Station	AR (%)	WT (s)	RT (s)
Group 1	4-2-7-17-16-15	70	313	2742	6-7-13-11	90	253	2673	5-1-10-12-9	100	244	2660
Group 2	7-1-17-12-16-11	86.67	252	3264	8-6-2-14-9-15-13	80	355	3688	5-4-10-16-15	80	189	3236
Group 3	7-1-17-16-15	93.33	297	4248	6-4-3-10-12-14-11	80	277	4368	5-2-1-9-13	83.33	309	3960
Group 4	8-4-7-16-15	100	282	3025	6-3-2-7-17-13-11	86.67	275	3309	5-1-12-9-15-13	100	296	3074
Group 5	5-6-7-1-17-16-13	96.67	252	3760	3-2-10-12	93.33	294	3424	5-8-4-9-15-13-11	86.67	296	3530

TABLE VII
RESULT OF CUSTOMIZED BUS SCHEME SET AT WTW = 10 min

Group Serial Number	Line1				Line2				Line3			
	Via station	AR (%)	WT (s)	RT (s)	Via station	AR (%)	WT (s)	RT (s)	Via station	AR (%)	WT (s)	RT (s)
Group 1	4-1-10-9-15	86.67	272	2776	2-6-13-16	73.33	283	2349	5-7-17-12-9	90	202	2746
Group 2	4-7-17-10-11	83.33	248	2991	8-6-1-12-14-9-13	73.33	279	3216	5-2-16-15	83.33	242	3079
Group 3	7-1-17-16-15	93.33	297	4248	6-2-1-14-13-11	76.67	259	4476	5-4-3-10-12-9	80	294	4569
Group 4	3-2-7-17-16-13	100	259	3081	6-1-12-13-11	80	284	3137	5-8-4-9-16-15	96.67	307	2760
Group 5	8-4-1-9-15-11	83.33	303	3796	6-3-2-10-12-13	100	234	3801	5-7-17-13	73.33	202	3358

TABLE VIII
RESULT OF CUSTOMIZED BUS SCHEME SET AT WTW = 20 min

Group Serial Number	Line1				Line2				Line3			
	Via Station	AR (%)	WT (s)	RT (s)	Via Station	AR (%)	WT (s)	RT (s)	Via Station	AR (%)	WT (s)	RT (s)
Group 1	4-2-16-15	80	276	2654	5-6-7-17-12-13-11	96.67	302	2630	5-7-1-10-12-9	96.67	315	2853
Group 2	4-7-17-16-11	83.33	254	3441	8-6-2-9-15-13	83.33	318	3841	5-2-1-10-12-14	86.67	277	3980
Group 3	4-7-1-17-10-16	100	369	4681	3-2-12-13	86.67	341	4373	5-6-1-14-9-15	83.33	269	4358
Group 4	4-7-17-16	96.67	349	3218	5-3-2-1-12-9-13	100	280	3087	5-8-5-7-15-13-11	100	315	3342
Group 5	8-6-7-1-17-9-13	100	373	3683	5-3-2-10-12-11	100	407	3813	5-8-4-15-13	100	292	3869

vehicles is 3, and the maximum passenger capacity of each vehicle is 30.

The hyperparameters of the environment, the neural network, and the RL framework in the model should be set. Among these hyperparameters, the reward discount factor is set to 0.9. The batch size is set to 64. Learning rate is set to 5×10^{-5}.

For each planned line, we choose the station where its status y_s^k is 1 as the result of the bus line planning scheme. Under the condition of WTW = 15 min, 15 lines are planned for passengers from 7:00 to 9:30 in the morning peak. Three indicators are used to measure the operational status of each line.

1) *Attendance Rate (AR)*: It is obtained by calculating the ratio of the number of onboard passengers carried to the vehicle capacity.
2) *Mean Waiting Time (WT)*: Passengers wait at the station for the vehicle to arrive, and it is smaller than WTW.
3) *Mean Running Time (RT)*: It refers to the time spent by passengers during vehicle operation, which excluding the waiting time. The running time of passengers is positively correlated with the running time of vehicles.

The mean travel time referred to in this article is the sum of the WT and the RT.

As shown in Table VI, the attendance rate of most lines is more than 80%, which means that passengers are relatively willing to accept customized bus service in this model. The mean running time of passengers who depart from 7:00 to 7:30 is about 45 min. Then, there will be two significant morning peaks, which respectively affect the customized bus passengers who depart from 7:30 to 8:30 and the customized bus passengers who depart from 9:00 to 9:30. Their mean travel time is over 55 min. The results show that congestion has an impact on the mean travel time of passengers.

Setting reasonable WTW is crucial for case studies, and few studies have explored the impact of WTW values on line optimization. In our study, we set WTW = 10 min and WTW = 20 min to generate schemes for the same case. As shown in Tables VII and VIII, the planning results of each line are different with different WTW. Compared to the condition of WTW = 15 min, the AR decreased by 3.55%, while the WT and RT decreased by 5.57% and 1.59%, respectively, on the condition of WTW = 10 min. The AR increased by 4.44%, while the WT and RT

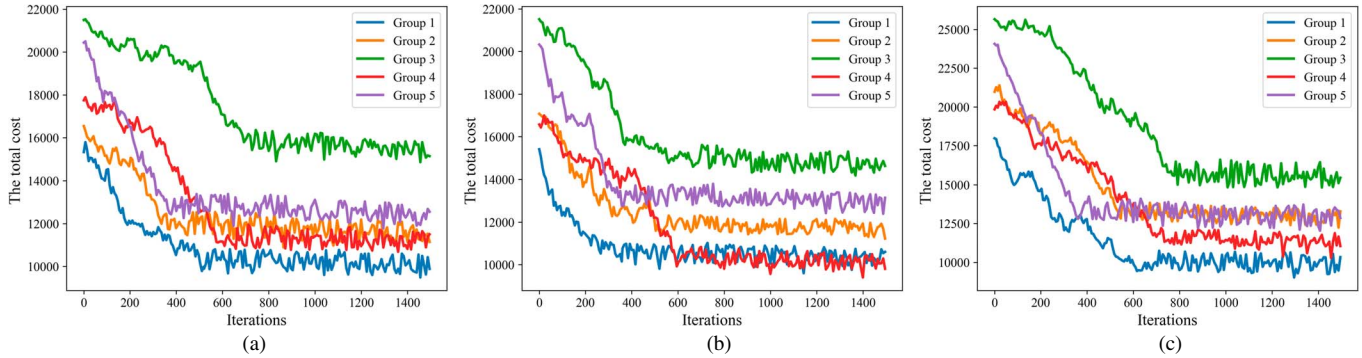


Fig. 7. Convergence curves of different groups on different WTW values. (a) WTW = 10 min. (b) WTW = 15 min. (c) WTW = 20 min.

increased by 13.29% and 5.32%, respectively, on the condition of WTW = 20 min. The impact of WTW on WT is significant.

Based on the above results, we can conclude that the WTW is set to 10 min is more suitable for case studies in super large cities, such as Beijing and New York, where passengers are not willing to spend more time to wait, which does not sacrifice too much attendance rate for operating companies. The WTW is set to 15 min is suitable for case studies in some medium-sized cities, such as Wuhan and Detroit, as well as for the city in this case study. The WTW is set to 20 min is not a good choice. Although it achieves a higher attendance rate, it will be a huge sacrifice for the time cost, and excessive waiting time will bring more uncertainty.

Fig. 7 shows the convergence trend of the total cost for different groups under different WTW values. It can found that the total reward for multiagent completing a training decreases continuously with the increase of iterations.

TABLE IX
COMPARISON OF COMBINATION ALGORITHMS

Station Planning Method	Line Optimization Model	AR (%)	Mean ± Std Value of WT (s)	Mean ± Std Value of RT (s)
K-means	ACO	92.0	385.31 ± 4.51	3945 ± 68.28
	GA	92.4	392.61 ± 2.81	3899 ± 39.38
	NSGA-II	**92.6**	369.18 ± 3.47	3820 ± 34.83
	MAQL	91.3	360.02 ± 1.94	3742 ± 30.1
	MADQN	92.0	331.85 ± 2.22	3571 ± 19.66
P-DN	ACO	89.5	366.39 ± 5.47	3857 ± 33.44
	GA	90.8	361.76 ± 4.71	3867 ± 40.04
	NSGA-II	89.5	346.46 ± 2.75	3795 ± 47.54
	MAQL	88.2	311.53 ± 2.75	3670 ± 37.78
	MADQN	90.2	302.91 ± 1.24	3546 ± 23.23
ST-DP	ACO	88.9	336.75 ± 2.65	3726 ± 51.86
	GA	88.6	342.83 ± 1.72	3792 ± 62.53
	NSGA-II	88.4	326.59 ± 3.01	3688 ± 34.63
	MAQL	87.7	312.09 ± 1.31	3579 ± 22.53
	MADQN	88.4	**278.93 ± 1.77**	**3397 ± 24.07**

C. Model Comparison

Compared with the commonly used algorithms for solving the customized bus problem, including two categories: station planning and line optimization.

The specific explanation and parameter settings are as follows:

For the station planning methods, we compare with two commonly used algorithms.

1) The K-means algorithm [13]: it is a clustering algorithm based on Euclidean distance, which does not exclude outliers. By setting the parameter K value, obtain K-class clusters. For this study, the parameter K is set to 9.
2) The P-DN algorithm [8]: It is an improved DBSCAN algorithm that allows the clustering results of the DBSCAN algorithm to be reclustered with a smaller ξ until all points are noisy. The P-DN algorithm only removes spatial outliers. For this study, the parameter ξ is set to 500 m, the parameter minPts is set to 10.

For the line optimization model, we compared with four commonly used algorithms.

1) *The improved ant colony optimization [13] (ACO)*: It is a probabilistic algorithm used to find optimal paths. The total number of pheromone is set to 100, the number of ants is set to 60, and the number of iterations is set to 200.
2) *The multiobjective genetic algorithm [33] (GA)*: It is a typical heuristic algorithm, which reflects the process of natural selection. The best individual is selected to produce the next generation of offspring. The PopSize is set to 100. The MaxGenes is set to 500. The parameter Pc is set to 0.8. The parameter Pm is set to 0.05.
3) *The three stage hybrid coding method based on NSGA-II algorithm [26] (NSGA-II)*: It is an elitist nondominated sorting genetic algorithm. The PopSize is set to 100. The MaxGenes is set to 200. The parameter Pc is set to 0.8. The parameter Pm is set to 0.1.
4) *The multiagent Q-learning [32] (MAQL)*: A value-based RL algorithm that builds a Q table from state and action to store Q values, and then selects the action that can obtain the maximum reward according to the Q value. The reward discount factor is set to 0.9.

The MAQL algorithm uses the same action space as our model to generate bus lines. Other algorithms directly generate bus lines. To better estimate the algorithm performance, the reported results of each line optimization algorithm are the mean and standard deviation values of ten independent experiments in Table IX. Also, the largest value and the lowest value have been deleted. The performance of different combination models is evaluated according to AR, WT, and RT. The best

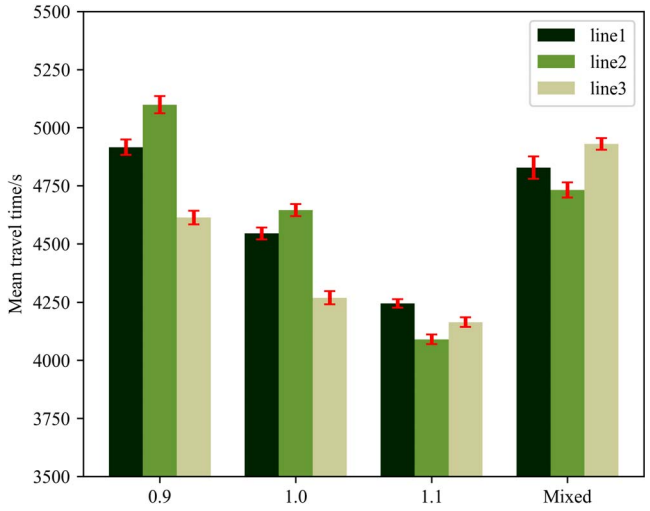


Fig. 8. Comparison of scheme with different running speed. The red line represents the standard deviation.

performing results are highlighted in bold. All results are generated under the condition of WTW = 15 min. As the K-means algorithm does not remove any noise points during station planning, the P-DN algorithm only removes a small number of spatial noise points. The number of passengers involved in line optimization will be more than the number of passengers generated by the ST-DP algorithm. The attendance rate of the first two algorithms will also be higher than that of the ST-DP algorithm on the same WTW value. Compared to our model, the combination model of K-means and NAGA-II increased the attendance rate by 4.2%, unfortunately, WT increased by 32.35% and MT increased by 12.45%. The relatively high attendance rate achieved by sacrificing huge time costs is not an ideal result, which means the presence of noise points has a significant impact on line optimization. Therefore, it is believed that the ST-DP algorithm is effective for station planning.

Under the same station planning conditions, the reinforcement learning algorithms can exhibit certain advantages, even the MAQL algorithm performs better than traditional heuristic algorithms. Our proposed algorithm (MADQN) can reduce WT (GA) by 18.3% and MT (ACO) by 10.4% compared to heuristic algorithms under the station planning results generated by K-means. The comparison shows that the performance of the combination model in this article is optimal and especially in reducing passenger waiting time.

D. Discussion

We select data from the testing set to evaluate the performance of the trained model. A feasible line may become suboptimal or infeasible when the speed changes [40]. Therefore, a well-performed bus line optimization method should meet the travel demand under different road conditions. Specifically, line optimization can significantly reduce the waiting time and running time of passengers during congestion by selecting appropriate stopping stations.

We scale the original driving speed according to the factors of 0.9, 1.1, and mixed speed to analyze the environmental sensitivity of our algorithm. The setting is to simulate different traffic scenes and different running speeds. For each case, we conducted ten experiments, respectively. Then, we calculated the mean and standard deviation of the mean travel time values of all passenger trips on different lines, as shown in Fig. 8. The travel time includes the waiting time and running time for each passenger.

We found that the mean travel time of passengers on the three lines increased by 8.16%, 9.77%, and 8.05%, respectively, at 0.9 times of the speed. We conclude that when the overall road congestion slows down the speed, the depth RL algorithm can provide an appropriate line planning scheme to reduce the impact of sudden changes. The travel time under different scaling factors changed slightly, decreasing by 6.6%, 11.94%, and 2.45%, respectively, at 1.1 times of the speed. The reason for this change is that when the overall line speed suddenly surges, the algorithm can get more rewards for obtaining a line with the running time that takes the least time. RL algorithm may sacrifice the waiting time of a few passengers to obtain the best running time of the line. In addition, the mixed speed has a significant impact on line optimization, and the mean travel time of passenger travel has increased by 6.22%, 1.87%, and 15.48%, respectively, which indicates that the depth RL algorithm can respond well to this change. The proposed algorithm can solve the challenge of vehicle running speed change by optimizing the bus operation line in real traffic scenarios.

VI. Conclusion and Future Work

In this article, we studied the station and line planning scheme of customized bus in urban areas. First, according to the spatiotemporal distribution characteristics of passenger travel demand, the ST-DP clustering algorithm is used to plan the customized bus joint station, and spatiotemporal outliers in the dataset are effectively removed. Second, a customized bus line planning model is established from the perspective of passenger travel time costs and operating costs. A multiagent RL method is designed to ensure effective training and response to incomplete information scenarios. Finally, the proposed optimization method and the designed solution strategy are verified by experiments on a real dataset, and 15 feasible lines are planned for potential customized bus groups on the condition of WTW = 15 min in Chongqing, China. Furthermore, we discussed the impact of different WTW values. Compared with the heuristic algorithm and the classical RL method, the proposed algorithm has lower social travel costs. This is because the algorithm proposed in this article can dynamically determine bus stations to meet different passenger demands and traffic state fluctuations. The research provides a new solution for bus companies to customize bus operation lines. The research results show that our method has practical benefits in improving bus operation efficiency and alleviating urban traffic congestion. In future research, we are actively accessing more real data to support the design of more scenarios and will apply the variable multiagent reinforcement learning method to solve the customized bus multiline planning problem under the environment of large-scale travel demand to achieve better social benefits.

REFERENCES

[1] Y. Li, X. Li, and S. Zhang, "Optimal pricing of customized bus services and ride-sharing based on a competitive game model," *Omega*, vol. 103, Sep. 2021, Art. no. 102413.

[2] Z. Ning et al., "Online scheduling and route planning for shared buses in urban traffic networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3430–3444, Apr. 2022.

[3] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14413–14423, Dec. 2020, doi: 10.1109/TVT.2020.3034800.

[4] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and J. Henry, "UAV trajectory planning in wireless sensor networks for energy consumption minimization by deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9540–9554, Sep. 2021, doi: 10.1109/TVT.2021.3102161.

[5] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding, and F. Shu, "Path planning for UAV-mounted mobile edge computing with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5723–5728, May 2020, doi: 10.1109/TVT.2020.2982508.

[6] X. Chen, Y. Wang, and X. Ma, "Customized bus line design model based on multi-source data," in *Proc. Int. Conf. Transp. Develop.*, Jul. 2018, pp. 218–228.

[7] Z. Han, Y. Chen, H. Li, K. Zhang, and J. Sun, "Customized bus network design based on individual reservation demands," *Sustainability*, vol. 11, no. 19, pp. 1–25, Jan. 2019.

[8] G. Qiu, R. Song, S. He, W. Xu, and M. Jiang, "Clustering passenger trip data for the potential passenger investigation and line design of customized commuter bus," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3351–3360, Sep. 2019.

[9] S. He and R. Song, "Micro-evolution algorithms for solving the dynamic location problem of customized bus stops," in *Theory, Methodology, Tools and Applications for Modeling and Simulation of Complex Systems*. Singapore: Springer, 2016, pp. 633–642, doi: 10.1007/978-981-10-2663-8_65.

[10] D. Huang, Y. Gu, S. Wang, Z. Liu, and W. Zhang, "A two-phase optimization model for the demand-responsive customized bus network design," *Transp. Res. C Emerg. Technol.*, vol. 111, pp. 1–21, Feb. 2020.

[11] C. Wang, C. Ma, and X. Xu, "Multi-objective optimization of real-time customized bus routes based on two-stage method," *Phys. Stat. Mech. Appl.*, vol. 537, Jan. 2020, Art. no. 122774.

[12] C. Shen, Y. Sun, Z. Bai, and H. Cui, "Real-time customized bus routes design with optimal passenger and vehicle matching based on column generation algorithm," *Phys. Stat. Mech. Appl.*, vol. 571, Jun. 2021, Art. no. 125836.

[13] W. Shu and Y. Li, "A novel demand-responsive customized bus based on improved ant colony optimization and clustering algorithms," *IEEE Trans. Intell. Transp. Syst*, vol. 24, no. 8, pp. 8492–8506, Aug. 2023.

[14] B. Yin, J. Li, and X. Wei, "Rational task assignment and path planning based on location and task characteristics in mobile crowdsensing," *IEEE Trans. Comput. Soc. Syst.*, vol. 9, no. 3, pp. 781–793, Jun. 2022, doi: 10.1109/TCSS.2021.3095946.

[15] X. Dou, Q. Meng, and K. Liu, "Customized bus service design for uncertain commuting travel demand," *Transp. Transp. Sci.*, vol. 17, no. 4, pp. 1405–1430, Dec. 2021.

[16] M. Behnke, T. Kirschstein, and C. Bierwirth, "A column generation approach for an emission-oriented vehicle routing problem on a multigraph," *Eur. J. Oper. Res.*, vol. 288, no. 3, pp. 794–809, Feb. 2021.

[17] F. Chen, H. Peng, W. Ding, X. Ma, D. Tang, and Y. Ye, "Customized bus passenger boarding and deboarding planning optimization model with the least number of contacts between passengers during COVID-19," *Phys. A*, vol. 582, Nov. 2021, Art. no. 126244.

[18] Y. Lyu, C.-Y. Chow, V. C. S. Lee, J. K. Y. Ng, Y. Li, and J. Zeng, "CB-planner: A bus line planning framework for customized bus systems," *Transp. Res. C Emerg. Technol.*, vol. 101, pp. 233–253, Apr. 2019.

[19] R. Guo, W. Zhang, W. Guan, and B. Ran, "Time-dependent urban customized bus routing with path flexibility," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2381–2390, Apr. 2021.

[20] Y. Wei, N. Jiang, Z. Li, D. Zheng, M. Chen, and M. Zhang, "An improved ant colony algorithm for urban bus network optimization based on existing bus routes," *ISPRS Int. J. Geo-Inf.*, vol. 11, no. 5, May 2022, Art. no. 5.

[21] Y. Yu, R. B. Machemehl, and C. Xie, "Demand-responsive transit circulator service network design," *Transp. Res. E Logist. Transp. Rev.*, vol. 76, pp. 160–175, Apr. 2015.

[22] Z.-M. Huang, W.-N. Chen, W. Shi, and X.-M. Hu, "Ant colony system for carpool service problem with high seating capacity," in *Neural Information Processing*. Cham, Switzerland: Springer, 2019, pp. 733–740, doi: 10.1007/978-3-030-36808-1_80.

[23] M. Gong, Y. Hu, Z. Chen, and X. Li, "Transfer-based customized modular bus system design with passenger-route assignment optimization," *Transp. Res. E Logist. Transp. Rev.*, vol. 153, Sep. 2021, Art. no. 102422.

[24] X. Chen, Y. Wang, and X. Ma, "Integrated optimization for commuting customized bus stop planning, routing design, and timetable development with passenger spatial-temporal accessibility," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2060–2075, Apr. 2021.

[25] J. Wang, Y. Cao, and Y.-H. Wang, "Customized bus route vehicle schedule method considering travel time windows," *China J. Highw. Transp.*, vol. 31, pp. 143–150, May 2018.

[26] C. Ma, C. Wang, and X. Xu, "A multi-objective robust optimization model for customized bus routes," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2359–2370, Apr. 2021.

[27] W. Joe and H. C. Lau, "Deep reinforcement learning approach to solve dynamic vehicle routing problem with stochastic customers," *Proc. Int. Conf. Autom. Plan. Sched.*, vol. 30, pp. 394–402, Jun. 2020.

[28] K. Manchella, M. Haliem, V. Aggarwal, and B. Bhargava, "PassGoodPool: Joint passengers and goods fleet management with reinforcement learning aided pricing, matching, and route planning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3866–3877, Apr. 2022.

[29] N. Peng, Y. Xi, J. Rao, X. Ma, and F. Ren, "Urban multiple route planning model using dynamic programming in reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8037–8047, Jul. 2022.

[30] Z. Wang, J. Yang, Q. Zhang, and L. Wang, "Risk-aware travel path planning algorithm based on reinforcement learning during COVID-19," *Sustainability*, vol. 14, no. 20, Art. no. 20, pp. 1–25, Jan. 2022 .

[31] Y. Geng et al., "Deep reinforcement learning based dynamic route planning for minimizing travel time," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2021, pp. 1–6, doi: 10.1109/ICCWorkshops50388.2021.9473555.

[32] A. Wang, H. Guan, P. Wang, L. Peng, and Y. Xue, "Cross-regional customized bus route planning considering staggered commuting during the COVID-19," *IEEE Access*, vol. 9, pp. 20208–20222, 2021.

[33] B. Zhang, Z. Zhong, Z. Sang, M. Zhang, and Y. Xue, "Two-level planning of customized bus routes based on uncertainty theory," *Sustainability*, vol. 13, no. 20, Jan. 2021, Art. no. 11418, doi: 10.3390/su1320.

[34] S. Han, H. Fu, J. Zhao, J. Lin, and W. Zeng, "Modelling and simulation of hierarchical scheduling of real-time responsive customised bus," *IET Intell. Transp. Syst.*, vol. 14, no. 12, pp. 1615–1625, 2020, doi: 10.1049/iet-its.2020.0138.

[35] J. Zhang, D. Z. W. Wang, and M. Meng, "Analyzing customized bus service on a multimodal travel corridor: An analytical modeling approach," *J. Transp. Eng. A, Syst.*, vol. 143, no. 11, Nov. 2017, Art. no. 04017057, doi: 10.1061/JTEPBS.0000087.

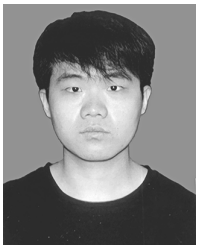
[36] L. Li, S. Wang, and F.-Y. Wang, "An analysis of taxi driver's route choice behavior using the trace records," *IEEE Trans. Comput. Soc. Syst.*, vol. 5, no. 2, pp. 576–582, Jun. 2018, doi: 10.1109/TCSS.2018.2831285.

[37] C. Lyu, X. Wu, Y. Liu, and Z. Liu, "A partial-Fréchet-distance-based framework for bus route identification," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9275–9280, Jul. 2022.

[38] X. Zhou, W. Liang, K. I.-K. Wang, and L. T. Yang, "Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations," *IEEE Trans. Comput. Soc. Syst.*, vol. 8, no. 1, pp. 171–178, Feb. 2021, doi: 10.1109/TCSS.2020.2987846.

[39] J. Ke, F. Xiao, H. Yang, and J. Ye, "Learning to delay in ride-sourcing systems: A multi-agent deep reinforcement learning framework," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2280–2292, May 2022, doi: 10.1109/TKDE.2020.3006084.

[40] Y. Wu, M. Poon, Z. Yuan, and Q. Xiao, "Time-dependent customized bus routing problem of large transport terminals considering the impact of late passengers," *Transp. Res. C Emerg. Technol.*, vol. 143, Oct. 2022, Art. no. 103859.



Wengang Li received the B.Eng. degree in information security from Chongqing University, Chongqing, China, in 2020. He is currently pursuing the Ph.D. degree in computer science with the Key Laboratory of Dependable Service Computing in Cyber–Physical Society of Ministry of Education, College of Computer Science, Chongqing University, Chongqing, China.

His research interests include intelligent transportation systems, data mining, and big data analysis.

Linjiang Zheng received the Ph.D. degree in computer science from Chongqing University, Chongqing, China, in 2010.

He is currently a Professor with Chongqing University, Chongqing, China. His research interests include spatiotemporal data mining, Internet of Things, and transportation Big Data analysis. He has authored or co-authored more than 50 papers. He has applied for more than 60 patents for invention, of which more than 30 have been authorized.

Dr. Zheng has won the second prize of the Science and Technology Progress Award of the Ministry of Education, second prize of the Chongqing Science and Technology Progress Award, and China Industry-University-Research Cooperation Innovation Achievement Award.

Longquan Liao received the bachelor's degree in Internet of Things engineering from the School of Computer Science, Chongqing University, Chongqing, China, where he is currently pursuing the Ph.D. degree with the Key Laboratory of Dependable Service Computing in the Cyber-Physical Society, which operates under the Ministry of Education.

His research interests encompass various fields within computer science, such as data mining, knowledge graph reasoning, graph representation learning, and intelligent transportation systems.

Xingze Yang received the B.Eng. degree in civil engineering from Lanzhou Jiaotong University, in 2018, and the master's degree in business information systems from Monash University, Melbourne, Victoria, Australia, in 2022.

His research interests include intelligent transportation systems, data analysis, and information systems.

Dihua Sun received the bachelor's degree from the Huazhong University of Science and Technology, Wuhan, China, in 1982, and the master's and Ph.D. degrees from Chongqing University, Chongqing, China, in 1989 and 1997, respectively.

He is currently a Professor with the School of Automation, Chongqing University. His research interests include cyber–physical systems, intelligent transportation system, computer-based control, data analysis, and decision support.

Weining Liu received the Ph.D. degree from Chongqing University, Chongqing, China, in 1999.

She is currently a Professor with the School of Computer Science and Technology, Chongqing University. Her research interests include distributed computing and services, e-commerce and modern logistics, RFID application technology, cyber-physical systems, and intelligent transportation system.