Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Spatiotemporal distance embedded hybrid ant colony algorithm for a kind of vehicle routing problem with constraints*#

Zhenhui FENG[1,2], Renbin XIAO[‡1,3]

[1]*School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China*
[2]*Institute of Artificial Intelligence, Huazhong University of Science and Technology, Wuhan 430074, China*
[3]*Key Laboratory of Image Information Processing and Intelligent Control, Ministry of Education, Wuhan 430074, China*
E-mail: feng_zh@hust.edu.cn; rbxiao@hust.edu.cn

Received Nov. 21, 2022; Revision accepted Apr. 6, 2023; Crosschecked May 11, 2023

Abstract: We investigate a kind of vehicle routing problem with constraints (VRPC) in the car-sharing mobility environment, where the problem is based on user orders, and each order has a reservation time limit and two location point transitions, origin and destination. It is a typical extended vehicle routing problem (VRP) with both time and space constraints. We consider the VRPC problem characteristics and establish a vehicle scheduling model to minimize operating costs and maximize user (or passenger) experience. To solve the scheduling model more accurately, a spatiotemporal distance representation function is defined based on the temporal and spatial properties of the customer, and a spatiotemporal distance embedded hybrid ant colony algorithm (HACA-ST) is proposed. The algorithm can be divided into two stages. First, through spatiotemporal clustering, the spatiotemporal distance between users is the main measure used to classify customers in categories, which helps provide heuristic information for problem solving. Second, an improved ant colony algorithm (ACO) is proposed to optimize the solution by combining a labor division strategy and the spatiotemporal distance function to obtain the final scheduling route. Computational analysis is carried out based on existing data sets and simulated urban instances. Compared with other heuristic algorithms, HACA-ST reduces the length of the shortest route by 2%–14% in benchmark instances. In VRPC testing instances, concerning the combined cost, HACA-ST has competitive cost compared to existing VRP-related algorithms. Finally, we provide two actual urban scenarios to further verify the effectiveness of the proposed algorithm.

Key words: Vehicle routing problem with constraints (VRPC); Spatiotemporal distance function; Labor division strategy; Ant colony algorithm (ACO)
https://doi.org/10.1631/FITEE.2200585 **CLC number:** TP399

1 Introduction

The route planning problem is an artificial intelligence research hotspot. The vehicle routing problem (VRP) has been one of the most widely studied route planning problems since it was proposed in the 1950s (Dantzig and Ramser, 1959). As the classic combinatorial optimization problem, VRP has been proved to be non-deterministic polynomial (NP) hard (Zhang WQ et al., 2020). In the past several decades, with the development of real logistics networks, there have been many different VRP variants that have been extensively studied (Jiang HW et al., 2022), such as the vehicle routing problem with time windows (VRPTW), capacitated vehicle routing problem (CVRP), and dynamic vehicle routing problem (DVRP). Most of these variants

‡ Corresponding author
* Project supported by the National Science and Technology Innovation 2030 Major Project of the Ministry of Science and Technology of China (No. 2018AAA0101200)
# Electronic supplementary materials: the online version of this article (https://doi.org/10.1631/FITEE.2200585) contains supplementary materials, which are available to authorized users
ORCID: Zhenhui FENG, https://orcid.org/0000-0001-7004-8868; Renbin XIAO, https://orcid.org/0000-0003-0951-2734

are built on logistics networks. However, the demand for route planning is no longer limited to logistics and distribution, but reflected in aspects such as personal travel (Zhang WY et al., 2022). In recent years, with the emergence of the Internet, big data, artificial intelligence, and other emerging technologies, traffic and travel methods have become more and more diverse. Shared travel methods such as shared rental cars and Internet car-hailing are gradually becoming a new trend in urban transport development (Jiang HZ and Zhang, 2019). How to combine various VRP problems with urban transportation and study the VRP in the car-sharing environment, is one of the directions with practical research significance at present.

The main feature of car-sharing is that users can make reservations online and submit their demand information to the platform. The operator then matches the vehicle to users, and schedules it to the user's neighborhood (Chapman et al., 2020). Compared to traditional modes of travel, car-sharing not only improves the utilization of resources and protects the ecological environment, but also brings a new travel experience to people, better meeting their need for convenience, comfort, and privacy (Hu et al., 2018). Currently, the service mode of car-sharing is becoming more and more refined and short term, which makes it more difficult for operators to manage and maintain (Chen DL et al., 2020). How to reasonably allocate vehicle resources to reduce costs as much as possible is the key to improving enterprise efficiency. Therefore, an effective vehicle scheduling plan must be developed to easily balance the supply and demand for shared vehicles. However, the existing car-sharing research has focused on station planning and incentive mechanisms, and less research has been conducted on vehicle scheduling problems for personal transportation travel. For example, Ma et al. (2019) studied the layout problem of charging stations and proposed an optimal allocation method for stations in view of the demand for shared electric vehicles. Nourinejad and Roorda (2014) proposed a user request response mechanism in shared car service based on an improved particle swarm optimization algorithm, which alleviated the imbalance between the supply and demand of a single program to a certain extent.

Compared with VRP issues, the vehicle scheduling problem in the car-sharing environment has more stringent spatiotemporal constraints and more complex objective functions. Considering the above characteristics, this study further investigates a kind of vehicle routing problem with constraints (VRPC) from the perspective of time and space constraints. VRPC is based on user orders, where each order has two positions: the starting point and the destination. In addition, the order restricts the vehicle usage time, which requires that the vehicle should perform the service within the user's acceptable waiting time.

VRPC is a variant of the VRP. The existing algorithms related to the VRP are inspiring for solving VRPC. The existing VRP solution approaches can be divided into two categories: exact algorithms and heuristic algorithms (Jiang HW et al., 2022). Exact algorithms establish the relevant mathematical model for a specific problem, and then employ mathematical methods to solve the model. In general, the exact algorithm can be solved to obtain the optimal problem solution. Common exact algorithms include the branch-and-bound method and dynamic programming method. Heuristic algorithms are another typical optimization algorithm that is generally able to give an approximate solution at an acceptable computational cost (Jia et al., 2022). There are many heuristic algorithms used for VRP solution, including the ant colony algorithm (ACO) and genetic algorithm (GA) (Xiao and Chen, 2023). According to the literature, these methods have been widely adopted and discussed. For example, some scholars (Beheshti and Hejazi, 2015) studied a kind of VRP with soft time windows, and proposed a hybrid column generation heuristic algorithm. Zhang JL et al. (2017) proposed a multi-objective co-evolutionary quantum algorithm for the time-dependent vehicle delivery path problem. Based on actual logistics processes, Sitek et al. (2021) proposed a hybrid constraint logic programming (CLP)-GA algorithm to solve the vehicle routing problem with simultaneous pick-up and delivery (VRPSPD). Wang Y et al. (2020) studied the periodic VRPTW, and designed a multi-objective simulated annealing (MOSA)-ACO algorithm for computational analysis.

Although the optimal solution to the VRP problem can be found using exact algorithms, the solving efficiency is obviously reduced when the problem scale increases, so we do not adopt this approach. Compared with exact algorithms, the heuristic algorithm based on swarm intelligence is a commonly used solution for NP-hard problems such as large-scale VRPs (Xiao et al., 2022). These algorithms are highly robust, simple to implement, and have parallel-style computation (Elsedimy and Algarni, 2022). The key to solving the problem is designing an optimization algorithm that could be adapted to the problem characteristics (Li et al., 2022). In VRPC, considering the flexibility of personal travel, the same vehicle might be scheduled many times within a short period. Compared with traditional vehicle scheduling problems, VRPC has more complex temporal and spatial constraints. Therefore, correctly addressing the spatiotemporal coupling and spatiotemporal conflicts between users, and precisely coordinating the allocation of individual vehicles to users, are the keys to rational planning of vehicle service routes. In addition, car-sharing is a service-oriented industry, whose emphasis is on user experience. Thus, the cost of user experience is an integral part of the actual cost.

In this study, the VRPC model is established by considering the operating cost and user experience, aiming at the lowest operating cost and highest user experience. Subsequently, to address the complex spatiotemporal characteristics of the problem, a spatiotemporal distance function is defined based on the reservation time and location information of the users. Compared to a single spatial distance, the spatiotemporal distance can better characterize and quantify the spatiotemporal differences between individual users. Finally, the spatiotemporal distance embedded hybrid ant colony algorithm (HACA-ST) is proposed and compared with other heuristics and CPLEX solvers to validate its effectiveness.

2 Problem formulation

This paper presents a study of a kind of VRPC in the car-sharing environment. In the VRPC model, users (or passengers) submit their travel information, which includes travel time, starting point, and destination, to the online platform in advance by making a reservation. This means that the platform operator can obtain the reservation information and serve the reserved users for the next time period at a specific moment. During the service, the vehicle serves each reserved user sequentially; the individual user uses the vehicle for a short period of time and the unit of charge is accurate to minutes and kilometers. The service ends for the user at the time the trip is completed; i.e., the use ends when the user reaches his/her destination (Chen DL et al., 2020).

2.1 Problem description

2.1.1 Time windows for reservations

Car-sharing is a new type of service industry based on the Internet and mobile terminals. In the actual scenario, a time limit is given when a user makes a reservation, and whether the user's demand can be satisfied within the reservation time period is a measurement of service quality and user experience. The reservation time period can be regarded as a time window function (Wang Y et al., 2018a). The time window is determined by the earliest time e and the latest time l of the user's reservation, and T_w is the acceptable waiting time for the user ($T_w=l-e$). No extra costs are incurred when the vehicle serves the user during the time window $[e, l]$; otherwise, extra costs are payable. If the vehicle arrives early at the starting point ($t<e$), it incurs parking costs and opportunity costs associated with the car being idle during the waiting period. If the vehicle arrives late ($t>l$), it results in a degraded user experience and user refunds, adding to the hidden costs (Sowmya and Sankaranarayanan, 2022).

2.1.2 Vehicle service process

During the vehicle service, both temporal and spatial attributes are considered, with each user order containing the location of the route and the time of service. The VRPC vehicle service process is shown in Fig. 1. As soon as the service starts, the vehicle serves each user in turn according to the scheduling solution. The total service time for user i can be expressed as the time between the earliest moment of the reservation and the moment of the user's arrival at the destination, during which the scheduling time (the time it takes for the vehicle to travel from another location to the user's starting point) and travel time (the time it takes for the vehicle to transport the user from the starting point to the destination) are experienced. After user i arrives at the destination, the vehicle is scheduled to the starting position of the next user j to continue providing service. The vehicle serves the

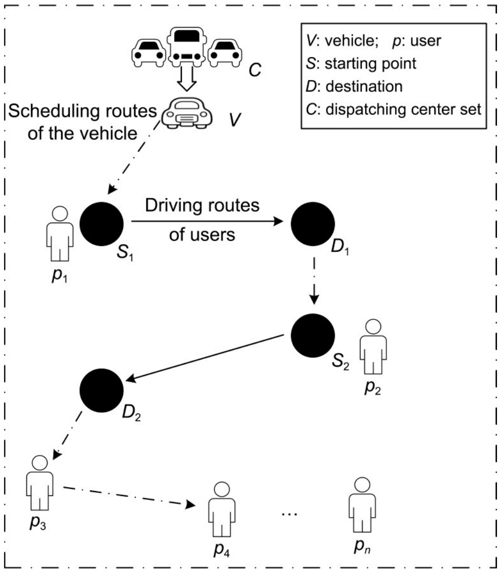

Fig. 1 Vehicle service process in the vehicle routing problem with constraints

different users sequentially until it cannot meet the time window of any user or the energy consumption reaches its availability limit; the scheduling then stops and the vehicle returns to the dispatching center. The above process is repeated until all users have been served to obtain the final number of vehicles and the scheduling routes of each vehicle (Alemi et al., 2019).

The goal of vehicle scheduling is to provide services to as many users as possible with limited vehicle resources under time and space constraints, to ensure that users are served within the time window as much as possible, and ultimately to obtain the scheduling solution with the lowest operating costs and the highest user experience.

2.2 Mathematical model

VRPC can be formulated as follows. A limit of $K_{\max}$ vehicles can be used to serve n users. Each user must be delivered from the starting point to the destination, and the service starting time is within the time window [e, l] of users as much as possible; otherwise, a certain penalty is applied. In addition, the mileage of each vehicle is limited to L. The ultimate goal is to find the optimal solutions that minimize the total cost, while meeting the needs of more users. The relevant symbols and definitions are shown in Table 1. Note that some reasonable assumptions must be followed for model establishment, which involve mainly information accuracy and energy consumption (see Section 1 of the supplementary materials for details).

$$\mathrm{Cost}_{\min} = C_{\mathrm{v}} + C_{\mathrm{s}} + C_{\mathrm{f}} + C_{\mathrm{t}}, \tag{1}$$

$$C_{\mathrm{v}} = Z_1 K, \tag{2}$$

$$\begin{aligned} C_{\mathrm{s}} = Z_2\Bigg[&\sum_{i\in P}\sum_{j\in P}\sum_{k\in[1,K]}\left(\mathrm{dist}\left(D_i, S_j\right)X_{ijk}\right) \\ &+\sum_{i\in P}\sum_{k\in[1,K]}\left(\mathrm{dist}\left(C, S_i\right)X_{cik}\right) \\ &+\sum_{i\in P}\sum_{k\in[1,K]}\left(\mathrm{dist}\left(D_i, C\right)X_{ick}\right)\Bigg], \end{aligned} \tag{3}$$

$$X_{ijk} = \begin{cases} 0, & \text{vehicle } k \text{ does not travel from } D_i \text{ to } S_j, \\ 1, & \text{vehicle } k \text{ travels from } D_i \text{ to } S_j, \end{cases} \tag{4}$$

$$\sum_{i\in P}\sum_{k\in[1,K]} X_{cik} = \sum_{i\in P}\sum_{k\in[1,K]} X_{ick} = K, \tag{5}$$

$$\sum_{k\in[1,K]}\sum_{i\in P\cup C} X_{ijk} = 1,\ \ \forall j\in P, \tag{6}$$

$$\sum_{i\in P\cup C} X_{ijk} = \sum_{i\in P\cup C} X_{jik},\ \ \forall j\in P, k\in[1,K], \tag{7}$$

$$C_{\mathrm{f}} = Z_3\sum_{i\in P}\mathrm{dist}\left(S_i, D_i\right), \tag{8}$$

$$\begin{aligned} X_{ijk}\sum_{i\in P}\sum_{j\in P}\Big[&\mathrm{dist}\left(D_i, S_j\right) + 0.5\Big(\mathrm{dist}\left(S_i, D_i\right) \\ &+\mathrm{dist}\left(S_j, D_j\right)\Big)\Big]\le L,\ \ \forall k\in[1,K], \end{aligned} \tag{9}$$

$$C_{\mathrm{t}} = \sum_{i\in P} F_i,\ \ \forall i\in P, \tag{10}$$

$$\begin{aligned} F_i = \theta_1 \max\left(e_i - T_{\mathrm{S}i}, 0\right) + \theta_2 \max\big(&T_{\mathrm{S}i} - \\ &l_i, 0\big),\ \ \forall i\in P, \end{aligned} \tag{11}$$

$$D_{1i} = T_{\mathrm{S}i} - e_i,\ \ \forall i\in P, \tag{12}$$

$$X_{jik}D_{1i}\le\gamma\left(l_i-e_i\right),\ \forall k\in[1,K], i\in P,\ j\in P\cup C, \tag{13}$$

$$X_{jik}\Big(T_{\mathrm{S}i} - \max\left(T_{\mathrm{D}j}, e_i\right) - T_{ji}\Big)\ge 0,\ \ \forall i,j\in P\cup C, \tag{14}$$

$$X_{jik}(T_{\mathrm{S}i} - T_{ji})\ge T_{\mathrm{g}ji},\ \ \forall i,j\in P\cup C, \tag{15}$$

$$T_{\mathrm{g}ji}\ge X_{jik}T_{\mathrm{D}j},\ \ \forall i,j\in P\cup C. \tag{16}$$

The model formulation is a mixed integer linear programming model with an objective function to minimize the total cost. As shown in Eq. (1), the total cost includes multiple costs. Eq. (2) represents the vehicle start-up cost. Eq. (3) represents the scheduling route driving cost of vehicles, where X_{ick} is a binary variable that represents whether vehicle k drives from D_i to the dispatching center, and c represents an element in the dispatching center set C. Similarly, Eq. (4) defines a binary variable X_{ijk}, which indicates the service

Table 1 Symbols and definitions of the VRPC model

Symbol	Description
P	User set, $\{p_1, p_2, \cdots, p_n\}$, indicates all users who have made a reservation in the next time period
C	Dispatching center set
S	Set of starting points, $\{S_1, S_2, \cdots, S_n\}$
D	Set of destinations, $\{D_1, D_2, \cdots, D_n\}$
$[e_i, l_i]$	Time window for service reservation for user p_i
K	Number of vehicles
Cost	Total cost, which consists of C_{v} (vehicle start-up cost), C_{s} (scheduling route cost), C_{f} (user traveling cost), and C_{t} (user experience cost)
Z	Cost coefficient, which consists of Z_1 (starting cost coefficient), Z_2 (unit cost of vehicle scheduling), and Z_3 (unit cost of user traveling)
dist(A, B)	Distance from A to B
X_{ijk}	Binary variable, indicating whether vehicle k travels from D_i to S_j
L	The maximum distance threshold for vehicle driving
F_i	Time penalty function of user p_i
$T_{\text{S}i}$	Trip starting time, i.e., the time when user p_i starts from the starting point
$T_{\text{D}i}$	Trip end time, i.e., the time point when user p_i arrives at the destination
$T_{\text{g}ji}$	Time point at which the vehicle leaves the destination of user p_j to the starting point of user p_i
$T_{\text{w}i}$	Acceptable waiting time of user p_i
θ_1	Unit parking cost incurred by the vehicle waiting for the user
θ_2	Unit delay costs incurred by service delays
γ	Time window limit coefficient
$D_{\text{l}i}$	Delay time, the time period between the user's reservation time point e_i and the trip starting time $T_{\text{S}i}$
T_{ij}	Scheduling time, the time when the vehicle is scheduled from the destination of the previous user p_i to the starting point of the next user p_j
$T_{\text{u}i}$	Traveling time, i.e., the time for user p_i to arrive at the destination from the starting point
V_{car}	Vehicle velocity, which can be used to calculate the arrival time of the vehicle

relationship between vehicles and users. Eq. (5) is the traffic flow constraint of the dispatching center, to ensure the balanced use of vehicles. The user service constraint is shown in Eqs. (6) and (7); i.e., each user can be served only once. Eq. (8) is the user traveling cost, which represents the energy consumption of driving the vehicle from the starting point to the destination. When the total driving distance of vehicle k (k=1, 2, $\cdots$, K) reaches threshold L, it needs to return to the dispatching center, as shown in inequality (9). Eq. (10) represents the user experience cost, which is related to the user reservation time and vehicle service time. When the time for the vehicle to reach the starting point of user p_i exceeds the time window $[e_i, l_i]$, the excess time carries a penalty, as shown in Eq. (11). Eqs. (12)–(16) are constraints related to the time dimension. Inequality (13) indicates that the delay time $D_{\text{l}i}$ of user p_i ($\forall i\in P$) shall not be greater than γ multiple of the time window. Inequalities (14) and (15) ensure that vehicle scheduling should be carried out within the time width allowed. Inequality (16) indicates that the vehicle can start scheduling to the starting point of user p_i only after the previous user p_j has arrived at the destination. The above is a brief description of the VRPC model, and a more detailed description can be found in Section 1 of the supplementary materials.

3 Spatiotemporal perspective on VRPC

Vehicle activities in the VRPC model have both spatial and temporal characteristics. The former refers to pairs of positions scattered within a particular region, i.e., the starting point and destination of each user, whereas the latter refers to the time window constraint imposed on the service provided to users. Because the position and state of a vehicle usually change over time, it would be reasonable to consider both spatial and temporal characteristics to represent its activity in an integrated approach.

Considering the multidimensional information of time and space, and to comprehensively measure the distance between all users, we refer to Qi et al. (2012) to describe the vehicle trajectories and user services of the VRPC model in the spatiotemporal coordinate system. Fig. 2 presents a simple case of VRPC, with two users and a viable vehicle service trajectory. The horizontal plane at the bottom denotes the two-dimensional (2D) space, and the vertical axis denotes time. The time window is represented as a cylinder with its height representing the width of the time window. From Fig. 2, it is clear that the vehicle serves two users p_1 and p_2, and the vehicle service route is $C\rightarrow S_1\rightarrow D_1\rightarrow S_2\rightarrow D_2\rightarrow C$. In the spatiotemporal coordinate system, the service route shows a cyclic ascending form $C\rightarrow S_1'\rightarrow D_1'\rightarrow S_2'\rightarrow D_2'\rightarrow C'$, which is named the space–time path, a basic notation in time geography. Fig. 2 depicts the ideal state of the vehicle space–time path; i.e., when the vehicle starts from C and arrives at user p_1's starting point S_1' and user p_2's starting point S_2', both of which are exactly within the reservation time window. However, for the two users p_i and p_j served by vehicles in sequence, assuming that the vehicle arrives at user p_i on time, the time T_{Sj} when it arrives at the starting point of the next user p_j should be as shown in Fig. 3, which can be divided into three cases: (1) $T_{Sj}<e_j$, the vehicle needs to wait for the user; (2) $e_j<T_{Sj}<l_j$, meeting the time window requirement of the user; (3) $T_{Sj}>l_j$, the user has to wait for the vehicle to arrive.

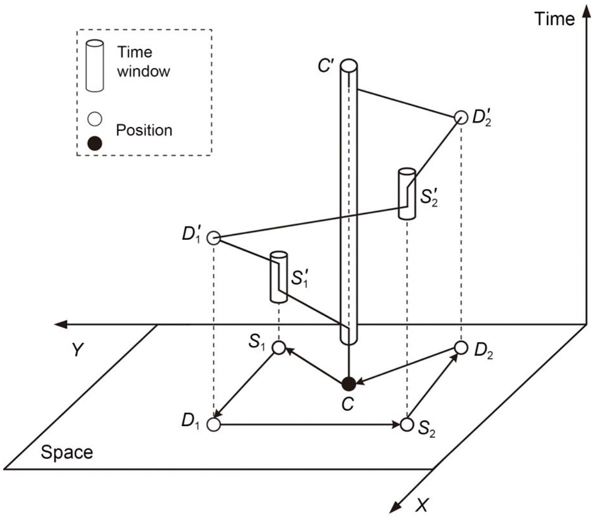

Fig. 2 Vehicle trajectory and user services of the vehicle routing problem with constraints

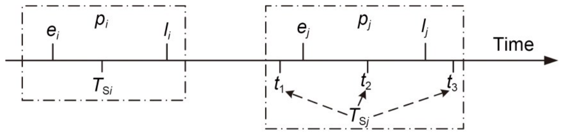

Fig. 3 Graph of time nodes among users

Accordingly, we define temporal distance functions distT_{ij} such as Eq. (17), where K_{t1} and K_{t2} are temporal distance coefficients, and $K_{t1}<K_{t2}$. As we know, the time window of each user in the actual scenario is independent. When we calculate the temporal distance between any two users, we should first sort the time windows and select the user with the previous time as the initial service object.

The spatial distance function distS_{ij} is shown in Eq. (19). Similarly, when we calculate the spatial distance, the user at the front of the time window should be selected as the initial service object.

$$\mathrm{dist}T_{ij}=\begin{cases}K_{t1}\left(e_j-T_{Sj}\right), & T_{Sj}<e_j,\\ 0, & e_j\le T_{Sj}\le l_j,\\ K_{t2}\left(T_{Sj}-l_j\right), & T_{Sj}>l_j,\end{cases}\quad \forall i,j\in P \text{ and } e_i<e_j, \tag{17}$$

$$T_{Sj}=0.5(e_i+l_i)+\left[\mathrm{dist}\left(S_i,D_i\right)+\mathrm{dist}\left(D_i,S_j\right)\right]\big/V_{\mathrm{car}},\quad \forall i,j\in P \text{ and } e_i<e_j, \tag{18}$$

$$\mathrm{dist}S_{ij}=\begin{cases}\mathrm{dist}\left(D_i,S_j\right), & e_i\le e_j,\\ \mathrm{dist}\left(D_j,S_i\right), & e_i>e_j,\end{cases}\quad \forall i,j\in P, \tag{19}$$

$$T\mathrm{dist}T_{ij}=\left(\mathrm{dist}\,T_{ij}-\mathrm{dist}\,T_{\min}\right)\big/\left(\mathrm{dist}\,T_{\max}-\mathrm{dist}T_{\min}\right),\quad \forall i,j\in P, \tag{20}$$

$$T\mathrm{dist}S_{ij}=\left(\mathrm{dist}\,S_{ij}-\min\left(\mathrm{dist}\,S\right)\right)\big/\left(\max\left(\mathrm{dist}\,S\right)-\min\left(\mathrm{dist}S\right)\right),\quad \forall i,j\in P, \tag{21}$$

$$\mathrm{dist\,ST}_{ij}=K_{\mathrm{T}}T\mathrm{dist}T_{ij}+K_{\mathrm{S}}T\,\mathrm{dist}\,S_{ij},\quad \forall i,j\in P. \tag{22}$$

As shown in Eqs. (20) and (22), the temporal distance dist$T$ and space distance dist$S$ are normalized respectively, and then the spatiotemporal distance distST is finally obtained, where K_{T} and K_{S} are the temporal and spatial weighting coefficients, respectively, obeying the following constraints: K_{T}, $K_{\mathrm{S}}\in(0, 1)$ and $K_{\mathrm{T}}+K_{\mathrm{S}}=1$.

4 Spatiotemporal distance embedded hybrid ACO

VRPC is a variant of the VRP, which is an NP-hard problem (Zhang WQ et al., 2020), and heuristic intelligent optimization algorithm is one of the commonly used solving techniques. In this study, an improved ACO approach is adopted to solve the VRPC problem. Compared with other meta-heuristic algorithms, the ACO algorithm achieves higher stability because of the positive feedback characteristics of the pheromone (Dang et al., 2022). However, the ACO algorithm has shortcomings, such as its tendency to be easily trapped in local optima and difficult to jump out (Wang Y et al., 2018a). Therefore, it is usually necessary to design and introduce other strategies for algorithm improvement to increase the convergence speed and avoid falling into local extremes. Earlier in this study, the spatiotemporal distance function was defined, which can more accurately represent the spatiotemporal relationship and constraint conflict among users. Based on the distance function and our previous work, we propose a spatiotemporal distance embedded hybrid ant colony algorithm (HACA-ST).

The HACA-ST algorithm procedure (Fig. 4) can be divided into two stages. The first stage is the user clustering process; a clustering method based on the spatiotemporal distance is designed, called the spatiotemporal clustering algorithm (STCA). The clustering method classifies user nodes according to the temporal and spatial attributes of their orders, which helps provide classification information for the second stage and construct high-quality feasible solutions. The second stage is a heuristic solution process, where the spatiotemporal clustering results and spatiotemporal distances are used as heuristic information, and an improved ant colony algorithm based on the mixed feedback mechanism (IMFACO) is proposed to optimize the service routes.

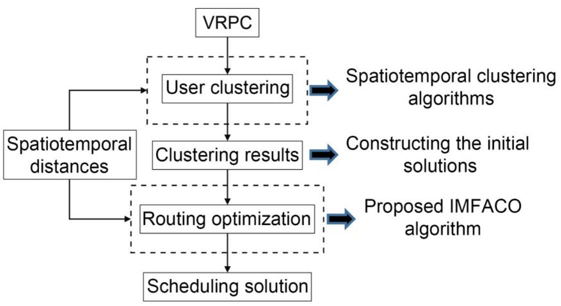

Fig. 4 Integrated framework of the spatiotemporal distance embedded hybrid ant colony algorithm

4.1 Spatiotemporal clustering algorithm

Clustering algorithms have been widely used to solve VRPs and their variants (Wang Y et al., 2018b, 2022; Brandão, 2020). In this study we propose a spatiotemporal clustering algorithm, which clusters users based on their time windows, coordinates, and other characteristics. Compared to clustering based on the spatial distance, the main measure of spatiotemporal clustering is the spatiotemporal distance function, to better measure the spatiotemporal differences between users. The K-means clustering algorithm is a common method for user clustering. Spatiotemporal clustering is based on the K-means clustering algorithm, in which the distance function is replaced with the spatiotemporal distance, and the order of the user reservation time windows within the cluster is referenced when selecting new clustering centers. The detailed process of the clustering algorithm is shown in Algorithm 1.

Algorithm 1 Spatiotemporal clustering algorithm (STCA)

Input: datasets P, S, and D, time window $[e, l]$, cluster number z, and maximum clustering number $\mathrm{Clu}_{\max}$
Output: set of clusters $C_1=\{C_{11}, C_{12}, \cdots, C_{1z}\}$
Data=$\{P_1, S_1, D_1, [e_1, l_1]; P_2, S_2, D_2, [e_2, l_2]; \cdots; P_n, S_n, D_n, [e_n, l_n]\}$
Randomly select data of z users from Data as cluster center Ct
repeat
  $C_{1i} = \phi \ (1\le i \le z)$ // Cluster C_{1i}
  for j=1, 2, ⋯, n **do**
    Calculate the spatiotemporal distance distST_{ji} between user P_j and each clustering center Ct_i
    Determine the cluster label of P_j based on the closest cluster center
    $\lambda_j=\mathrm{argmin}_{i\in\{1,2,\cdots,z\}}\mathrm{distST}_{ji}$
    Classify user P_j into the corresponding cluster $C_{1\lambda_j}=C_{1\lambda_j}\cup\{P_j\}$
  end for
  for i=1, 2, ⋯, z **do**
    Sort the internal users of each cluster C_{1i} according to the middle moments of the time window $(e+l)/2$
    Select the user whose $(e+l)/2$ lies in the middle as the new cluster center Ct_i
  end for
Until the current cluster center Ct does not change or the maximum clustering number $\mathrm{Clu}_{\max}$ is reached

To describe the algorithm more clearly, an example of the clustering process is given in Section 2.1 of the supplementary materials. Furthermore, in the process of STCA execution, the value of cluster number z should be related to the total number of orders. In this study, z is determined based on the temporal distance dist$T$ and travel distance thresholds L between users in the same time period. Detailed steps are illustrated as follows:

1. The average acceptable waiting time $\widetilde{T}_{\mathrm{w}}$ of the users is calculated, and the users are grouped with $\widetilde{T}_{\mathrm{w}}$ as the time interval. The maximum number of time distances between users within the group that are greater than γ times the acceptable time $\widetilde{T}_{\mathrm{w}}$ is counted and set as the initial cluster number z.

2. The clustering is executed according to the temporal distance between users, following the steps in Algorithm 1. At the end of clustering, if the total distance (the sum of the user traveling distance and vehicle scheduling distance) within a cluster is greater than the threshold L, the value of z needs to be increased by making z=z+1.

3. The above process is repeated until the time and distance constraints are satisfied, and the final cluster number z is obtained.

4.2 Initial solution generation strategy

We propose a cluster-first route-second heuristic approach (i.e., HACA-ST), to solve a kind of VRPC. According to the integrated framework of HACA-ST, before performing route optimization, the clustering results of the first stage should be fully used to generate the initial route sets.

In this study, a heuristic semi-random strategy is proposed to construct the initial solution set. First, the number of initial solutions is determined, and then the initial solutions are generated according to the clustering results. The pseudocode is shown in Algorithm 2. Each group (cluster) in the clustering result is randomly selected as either a sequentially ranked object or a randomly ranked object. If a cluster C_{1i} is a sequentially ranked object, the users in the group are sorted according to the time window order. Similarly, if it is a randomly ranked object, the users in the group would be randomly sorted. An initial solution S_{1i} can be obtained by combining all the ranked objects in random sequences Randperm. The above operation steps are repeated cyclically until the initial solution set satisfying the quantity requirement is generated.

Algorithm 2 Heuristic semi-random strategy

Input: clustering results C_1={C_{11}, C_{12}, $\cdots$, C_{1z}},
and the number of initial solutions N

Output: initial solution set S_1={S_{11}, S_{12}, $\cdots$, S_{1N}}

Set count=0
while count <N **do**
  for i=1, 2, $\cdots$, z **do**
    Cluster C_{1i} randomly selected to become a sequentially ranked object or a randomly ranked object
    if C_{1i}∈Sequentially ranked object
      Sort the users of C_{1i} according to the time window order and obtain Part_i=sort{C_{1i}}
    else
      Sort randomly the users in cluster C_{1i} and obtain Part_i=rand{C_{1i}}
    end if
  end for
  Generate a random sequence Randperm of [1: z]
  Combine all part objects based on Randperm to generate the initial solution $S_{1\mathrm{count}}$
  count=count+1
end while

To describe the solution generation process more clearly, an example based on the algorithm steps is given in Section 2.2 of the supplementary materials for a detailed description.

4.3 IMFACO algorithm

Because of the positive feedback based search method (Bonabeau et al., 2000), ACO has the disadvantage of slow convergence and a tendency to fall into local minima. In response to the above shortcomings, our team previously proposed a kind of extended ant colony algorithm based on the mixed feedback mechanism (MFACO), and achieved better results (Feng and Xiao, 2022). In this study, we further improve MFACO and propose an IMFACO for routing optimization of VRPC. This algorithm introduces the spatiotemporal clustering results and spatiotemporal distance functions as heuristic information. To help the reader better understand the IMFACO algorithm, we introduce the ACO algorithm briefly in Section 2.3 of the supplementary materials.

The IMFACO algorithm defines an extended ant with global search capability based on the traditional

ACO, and proposes a multi-role balancing mechanism, so the algorithm contains both a positive pheromone feedback search and a dynamic balancing mechanism with negative feedback characteristics to ensure global search capability. In addition, the algorithm improves the pheromone update mechanism based on the stimulus-response model, which also further accelerates the convergence of the algorithm. As shown in Fig. 5, the flow of the IMFACO algorithm based on the spatiotemporal distance can be divided into four parts, namely role division, path construction, pheromone updating, and multi-role balancing mechanism.

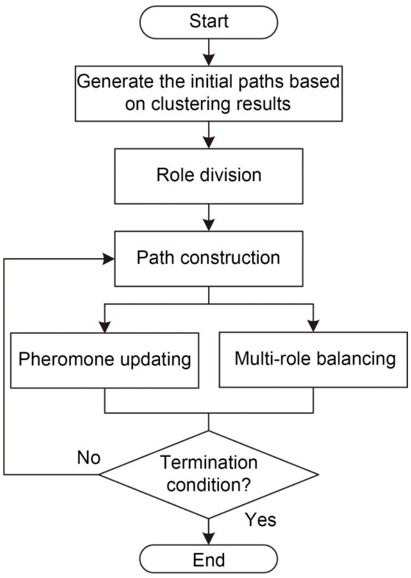

Fig. 5 Framework of the improved ant colony algorithm based on the mixed feedback mechanism

4.3.1 Role division and path construction

In the traditional ACO, all individual ants follow the same search rules and rely on pheromone concentration information to construct paths. Based on the traditional ACO, we introduce the extended ants in addition to the regular ants of the pheromone positive feedback search. As shown in Fig. 6, the population is assumed to have a total of m individual ants, with m_1 regular ants and m_2 extended ants. The paths of the regular ants are constructed in a manner similar to the ACO algorithm, based on pheromone concentration information for probabilistic transfer, but the heuristic information $\eta_{ij}(t)$ is set as the inverse of the spatiotemporal distance distST_{ij} between the two nodes (i, j). Instead of relying on pheromones to select paths, the extended ant selects existing paths for combinatorial exchange and constructs new paths based on information from known solutions.

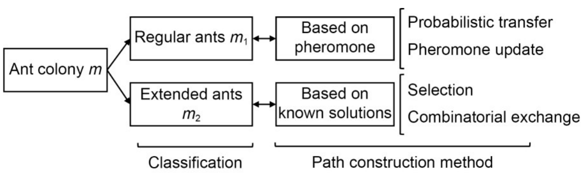

Fig. 6 Role division of ant colony

At the beginning of each generation of the ant population search, all regular ants construct m_1 paths as known solutions (i.e., existing paths) based on pheromone positive feedback. Subsequently, based on the proportional selection strategy and the roulette wheel method, the required m_2 paths are selected and allocated to each extended ant. The selection probability P_{bi} of each path is proportional to the inverse of the total length L_{ti} of the path, to ensure that the extended ants preferentially choose the path with smaller length. For the VRPC model, the total length L_{ti} can be considered as a fitness function value calculated according to the objective function Cost.

After the selection is completed, each extended ant randomly selects an object among the other extended ants except itself for combinatorial exchange, and constructs a new path. The method of combinatorial exchange is designed by referring to the two-point crossover operator in GA, which could cause the extended ants to have a strong global search capability. Thus, path combination exchange makes it easier for the extended ants to jump out of local extremes and ensures the global search ability. The detailed procedure of combinatorial exchange is consistent with that of IMFACO, which can be found in our previous work (Feng and Xiao, 2022). According to the extended ant's path construction method, the extended ant does not rely on pheromone information, but rather on individual selection and combinatorial exchange based on existing paths. Therefore, the diversity of the population is ensured by role division, and the algorithm can effectively jump out when it falls into local extremes and avoid premature convergence.

4.3.2 Multi-role balancing mechanism

As shown in Fig. 7, regular ants construct paths with reference to pheromone concentration, which has

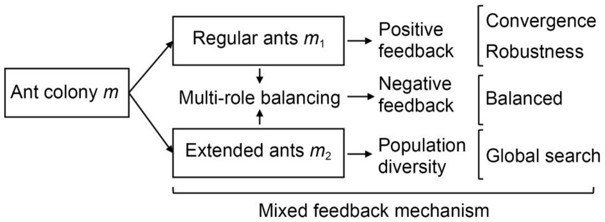

Fig. 7 Mixed feedback mechanism

positive feedback characteristics and ensures effective convergence and robustness of the algorithm. Extended ants construct new paths through existing paths, which exhibits strong global search ability and ensures the diversity of the population. The role-balancing mechanism dynamically adjusts the proportion of regular ants and extended ants in the colony, balancing the stable convergence and global search ability of the algorithm. The mechanism is designed based on the stimulus-response model. In essence, it is a regulation method in the negative feedback mode.

When a population is constructing a set of paths, the more paths all ants take and the less similar the paths are, the more diverse the population is. Conversely, as the population gradually focuses on one or more paths, the higher the similarity, the weaker the population diversity. In the proposed IMFACO algorithm, different tasks (exploitation or exploration) are performed to balance population diversity, as shown in Fig. 8. In summary, the role-balancing mechanism is a negative feedback adjustment method that adjusts individual roles according to population diversity. Thus, the role-balancing mechanism is designed based on the stimulus-response model (Xiao and Wang, 2018); the details can be found in our previous work (Feng and Xiao, 2022).

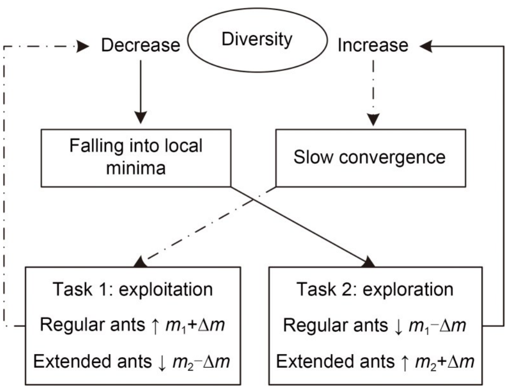

Fig. 8 Multi-role balancing mechanism

4.3.3 Pheromone updating

According to the population evolution principle and biological genetic properties, when previous generation populations are replaced by a new generation population, the delivery of useful heuristic information to the offspring individuals for guidance can help the population retain better characteristics and accelerate the convergence of the algorithm. Therefore, we introduce the elite strategy (Lin and Kernighan, 1973) and design a pheromone update mechanism based on the stimulus-response model. According to the ACO pheromone update rule, although the extended ants do not use the pheromone as search information, it still participates in the information system update when passing through the node paths.

The design principle of the pheromone update mechanism is as follows: (1) First, each ant has its own pheromone update threshold, which is related to the corresponding path length (i.e., fitness function value) of individual ants. (2) The external environment has a corresponding stimulus, and the stimulus value is related to the value of the average fitness function value (i.e., the average path) of the population. (3) The probability of updating pheromone is related to the individual threshold and the external environment stimulus value, which follows the stimulus-response model.

For details of the pheromone updating formulation, please see our previous work (Feng and Xiao, 2022). Generally speaking, the larger the path length, the poorer the quality of the corresponding solution, and the lower the probability of its pheromone update. The smaller the path length, the better the quality of the corresponding solution, and the higher the probability of its pheromone update. Therefore, in the initial stage of algorithm operation, the quality gap between feasible solutions is large, and high-quality feasible solutions will update the pheromone with a higher probability, which ensures the rapid accumulation of pheromone in the initial stage and accelerates the convergence of the algorithm. However, the algorithm does not completely discard inferior solution information, and solutions with poor quality also have a certain probability of pheromone update, which avoids the problem of algorithm stagnation to a certain extent.

The preceding description is the main content of the IMFACO algorithm. In addition, to further strengthen the local search capability, the algorithm introduces the 2-opt exchange to perform local tuning. The detailed steps of the whole algorithm (HACA-ST) can be found in Section 4.4.

4.4 HACA-ST algorithm procedure

The HACA-ST algorithm flowchart (Fig. 9) has two parts: STCA and IMFACO.

In the IMFACO part, the ant population consists of regular ants and extended ants. During the initialization of the algorithm, the parameters to be initialized include the population size m, the percentage of the two types of ants, the maximum number of iterations to be performed by the local search N_c, and ACO-related parameters such as the pheromone intensity factor Q, pheromone volatility factor ρ, pheromone matrix τ, heuristic factors α and β, and maximum number of iterations N_{max}. The regular ants use the spatiotemporal distance instead of the spatial distance, and construct paths according to the node probability selection formula. Extended ants select the existing paths to combine and exchange them, and construct new paths based on the information of known solutions. The fitness evaluation index of all ant paths is calculated according to the objective function of the VRPC model. In addition, if the model constraints cannot be satisfied between neighboring nodes in a path, the service of the current vehicle is terminated and returned to the dispatching center, and a new vehicle is dispatched to continue the subsequent path nodes until all user orders are served.

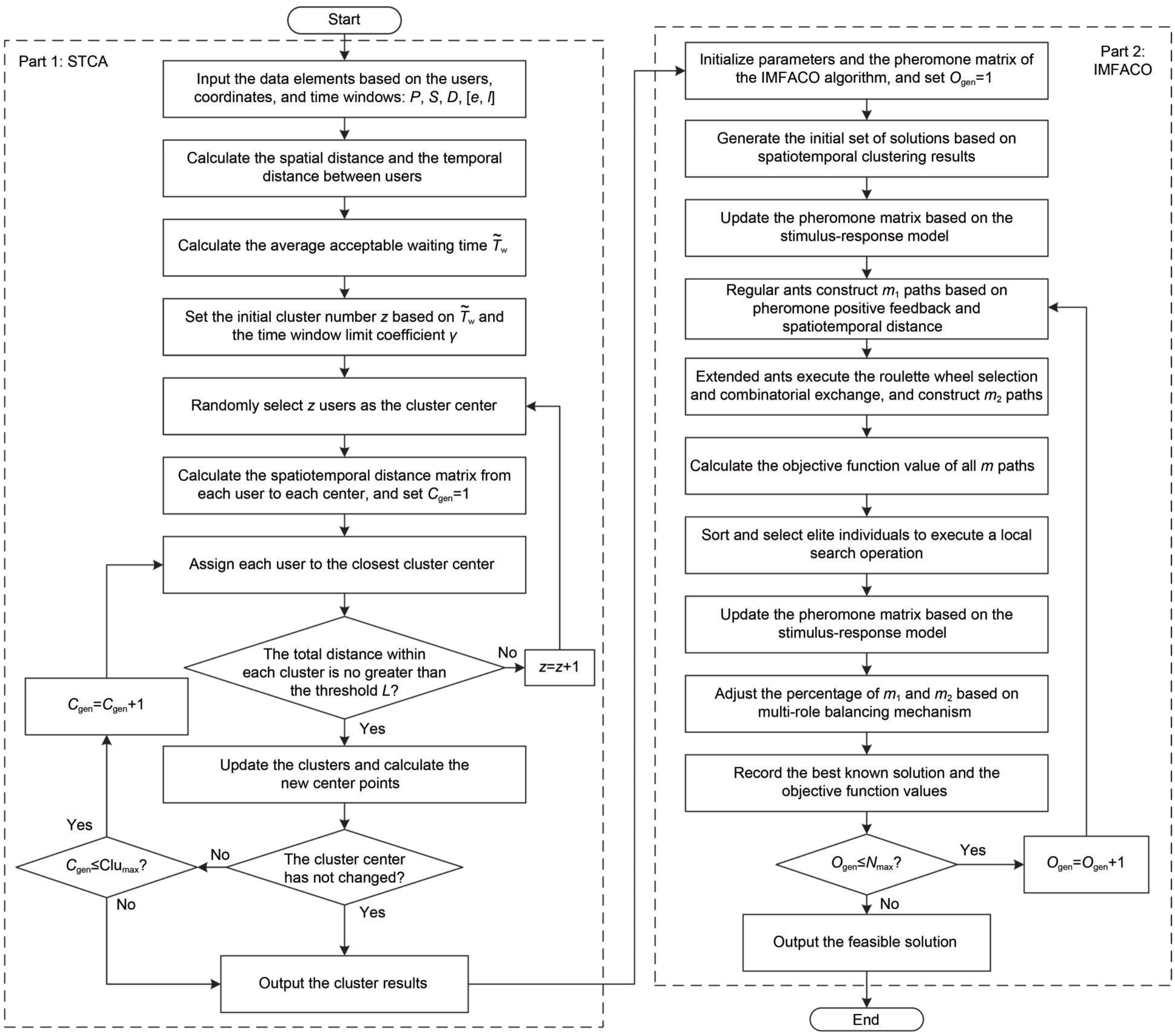

Fig. 9 Detailed flowchart of the HACA-ST algorithm

5 Numerical examples

5.1 Experimental and parameter setting

The VRPC model could be regarded as an extension of the classical VRP. Therefore, to test the accuracy and stability of the algorithm solution, our testing instances are divided into two groups. The first group includes the benchmark VRP instances to test the performance of the algorithm on generic instances. The second group includes the VRPC testing instances; this group is designed based on the benchmark instances, to compute and analyze the VRP in the car-sharing environment proposed in this study. In the testing instances, the HACA-ST algorithm is compared with other optimization algorithms and known optimal solutions to verify the effectiveness of the algorithm. In addition, to further assess the solution performance of the algorithm in a realistic environment, several scenarios are designed based on the urban area of Wuhan city. The detailed settings of the test instances and scenario instances are described below.

In this study, the algorithms are implemented based on MATLAB 2017b. The processor is Intel® Core™ i7-7500U, the main frequency is 2.7 GHz, and the memory is 16 GB RAM. The parameters of the HACA-ST algorithm are set as follows: α=2, β=5, ρ=0.6, Q=100, m_1=0.8m, m_2=0.2m, m=2C_{num} (C_{num} is the number of coordinates), $K_{\mathrm{t}1}$=1, $K_{\mathrm{t}2}$=5, $\mathrm{Clu}_{\max}$=100, $N_{\max}$=1000, and N_C=n. The termination conditions for all algorithms are set as follows: (1) One thousand iterations are completed; (2) There are no more enhancements that can be made over the minimum solution within 50 consecutive iterations for each run. All algorithms are used to perform 10 separate runs for each instance, and the optimal values can be obtained from these 10 runs.

5.2 Results of testing instances

5.2.1 VRPTW benchmark instances

To assess the quality of the HACA-ST algorithm, we first use the Solomon benchmark VRPTW instances for testing (Solomon, 1987), which come in three sets *r*, *c*, and rc, classified with respect to the geographical locations of the nodes. It is important to illustrate that when using the algorithm to solve these benchmark instances, the objective function is to find a shortest path according to the VRPTW model (Errico et al., 2018).

In this study, we compare the results of the proposed HACA-ST algorithm with those of other optimization algorithms, including ACO (Dorigo and Gambardella, 1997), ant algorithm based on direction-coordinating (AADC) (Meng et al., 2013), improved wolf pack algorithm (IWPA) (Chen YY et al., 2022), and improved genetic algorithm (IGA) (Wang WJ, 2010). The comparison results are shown in Table 2, and 15 benchmark instances are selected for testing. The best

Table 2 Comparison results of benchmark instances

Instance	Best known value	Minimum					Gap (%)				
		HACA-ST	ACO	AADC	IWPA	IGA	HACA-ST	ACO	AADC	IWPA	IGA
C101	827.3	828.9	900.5	875.3	853.6	870.1	0.19	8.85	5.80	3.18	5.17
C102	827.3	828.9	929.1	894.8	832.7	890.4	0.19	12.31	8.16	0.65	7.63
C103	826.3	828.1	881.7	900.5	859.8	878.6	0.22	6.70	8.98	4.05	6.33
C104	822.9	824.7	907.5	849.1	893.2	892.6	0.22	10.28	3.18	8.54	8.47
C105	827.3	828.9	896.8	913.0	836.7	889.3	0.19	8.40	10.36	1.14	7.49
C201	589.1	594.2	673.1	639.3	619.2	652.6	0.87	14.26	8.52	5.11	10.77
C202	589.1	591.5	656.1	627.1	620.9	637.2	0.41	11.37	6.45	5.40	8.17
C203	588.7	591.1	653.2	636.9	606.2	655.1	0.41	10.96	8.19	2.97	11.28
C204	588.1	590.6	674.5	645.9	609.9	631.4	0.43	14.69	9.83	3.71	7.35
C205	586.4	588.9	664.9	623.4	620.2	620.2	0.43	13.39	6.31	5.76	5.76
R101	1637.7	1650.8	1822.3	1771.8	1769.0	1809.7	0.80	11.27	8.19	8.02	10.50
R102	1466.6	1486.1	1664.7	1567.9	1560.9	1629.7	1.33	13.51	6.91	6.43	11.12
R103	1355.3	1377.1	1462.8	1455.3	1381.4	1434.2	1.61	7.93	7.38	1.93	5.82
RC101	1619.8	1696.9	1851.1	1786.2	1785.5	1777.8	4.76	14.28	10.27	10.23	9.76
RC104	1132.3	1135.4	1244.2	1192.7	1187.9	1273.9	0.27	9.88	5.33	4.91	12.51

known value is compiled from Yu et al. (2022). The Gap values in the table are the gaps between the best known value and the minimum value. It is clear that the results obtained by HACA-ST are close to the best known solution. HACA-ST is more accurate than other algorithms, reducing the smallest route length by 2%–14%.

5.2.2 VRPC testing instances

To assess the performance of the HACA-ST algorithm in solving VRPC problems, we randomly select eight groups of different-scale instances from datasets C101, R101, RC101, and C1_2_1. Each instance contains a vehicle dispatching center and a number of user reservation trips, with the coordinate data selected from the above datasets as the starting point and destination for each user.

These VRPC testing instances are divided into three different scale levels. The first three groups are small-scale instances, including 30 users (i.e., 60 coordinates). The middle three groups are medium-scale cases, including 60 users (i.e., 120 coordinates). The last two groups are large-scale cases with n=100, including 100 users (200 coordinates). The specific parameters of the VRPC model are set as shown in Table 3. The comparison results are shown in Table 4. The deviations (i.e., Dev) in the table are the gaps between the average value and the minimum value, which can represent the stability of the algorithm to some extent (the smaller the deviation, the more stable the algorithm). From the results, it can be seen that among all instances, the optimal and average values obtained by HACA-ST are the best, with the smallest deviation and the highest stability.

Table 3 Parameters of the VRPC model

Parameter	Value	Parameter	Value
Z_1	100 CNY	γ	1.5
Z_2	0.98 CNY/km	L	200 km
Z_3	0.48 CNY/km	θ_1	1 CNY/min
V_{car}	30 km/h	θ_2	7.5 CNY/min

In addition, VRPC is essentially a variant of VRP. To better assess the effectiveness of the algorithm, we refer to existing VRP-related algorithms, including the strength pareto evolutionary algorithm (SPEA) (Baños et al., 2013) and improved ant colony algorithm (IACO) (Yang et al., 2019), modify these algorithms to make them applicable to the VRPC problem, and compare them with the HACA-ST algorithm. In this

Table 4 Comparison results of VRPC test instances

Instance	Algorithm	Minimum	Average	Dev (%)	Instance	Algorithm	Minimum	Average	Dev (%)
C1 n=30	ACO	1074.7	1176.6	9.48	C5 n=60	ACO	2232.4	2516.6	12.73
	AADC	922.4	1050.9	13.94		AADC	2182.5	2441.4	11.86
	IWPA	891.5	983.9	10.36		IWPA	2201.8	2509.4	13.97
	IGA	947.4	1068.3	12.76		IGA	2182.5	2431.6	11.41
	HACA-ST	879.4	897.2	2.02		HACA-ST	2087.2	2170.5	3.99
C2 n=30	ACO	1097.3	1279.1	16.57	C6 n=60	ACO	1999.6	2306.8	15.36
	AADC	921.0	1026.3	11.44		AADC	1950.2	2217.2	13.69
	IWPA	1004.3	1127.7	12.29		IWPA	1903.4	2138.8	12.37
	IGA	988.6	1135.8	14.89		IGA	2067.2	2402.6	16.22
	HACA-ST	900.9	920.1	2.13		HACA-ST	1854.8	1929.5	4.03
C3 n=30	ACO	868.1	970.5	11.80	C7 n=100	ACO	3564.0	4205.9	18.01
	AADC	740.8	820.2	10.72		AADC	3434.4	3909.0	13.82
	IWPA	718.7	785.3	9.26		IWPA	3500.4	4072.8	16.35
	IGA	791.2	868.9	9.82		IGA	3564.0	4126.8	15.79
	HACA-ST	697.1	723.1	3.73		HACA-ST	3341.3	3553.8	6.34
C4 n=60	ACO	1674.1	1839.5	9.88	C8 n=100	ACO	3594.7	4236.8	17.86
	AADC	1560.9	1670.4	7.02		AADC	3462.2	3959.1	14.35
	IWPA	1583.4	1725.7	8.99		IWPA	3510.4	3922.5	11.74
	IGA	1652.3	1821.7	10.25		IGA	3510.4	4089.7	16.50
	HACA-ST	1526.1	1577.2	3.35		HACA-ST	3307.5	3493.0	5.61

study, C1, C4, and C7 for three scale test instances are selected for experimental comparison, and the comparison results are shown in Table 5. The HACA-ST algorithm also has higher accuracy and stability, reducing the total cost by 2%–9%.

Table 5 Comparison results of the proposed algorithm and two other algorithms

Ins	Algorithm	Min		Avg		Dev
		Value	Diff (%)	Value	Diff (%)	(%)
C1	IACO	897.6	2.07 ↑	924.3	3.02 ↑	2.89
	SPEA	891.5	1.38 ↑	924.4	3.03 ↑	3.55
	HACA-ST	879.4		897.2		2.02
C4	IACO	1560.9	2.28 ↑	1694.8	7.46 ↑	8.58
	SPEA	1576.3	3.29 ↑	1691.9	7.27 ↑	7.33
	HACA-ST	1526.1		1577.2		3.35
C7	IACO	3449.7	3.24 ↑	3785.0	6.51 ↑	9.72
	SPEA	3460.5	3.57 ↑	3872.6	8.97 ↑	11.91
	HACA-ST	3341.3		3553.8		6.34

Ins: instance; Min: minimum; Avg: average; Dev: deviation; "Diff (%)" and " ↑ " indicate the percentage increase in cost

5.3 Location setting of real scenarios

Wuhan is one of the transportation hubs in the central region of China, and can be regarded as a suitable research subject for this study. A real traffic network in Wuhan is used to test the applicability and effectiveness of the proposed methods.

We use the open-source data from OpenStreetMap to set up the scenario instances. Pairs of points of information (POIs) are selected sequentially, and set as the user's origin and destination. The selected areas are limited in longitude 113° 96′–114° 60′E and latitude 30°31′–31°40′N. After coordinate transformation, the area can be roughly expressed as a rectangular area of 60 km×120 km. In addition, a random normal distribution rule is used to generate the service appointment times, and the coordinate values and time points are rounded for ease of representation and calculation.

The scenario instances are divided into two groups, case 1 (including 25 users and 1 vehicle dispatching center) and case 2 (including 30 users and 2 vehicle dispatching centers). Fig. 10 shows the distribution of users for the scenario instances. The circle shape indicates the vehicle dispatching center, the star symbol is the user's starting point, and the teardrop symbol indicates the user's destination. Instance datasets are available in Section 3 of the supplementary materials.

5.4 Results of scenario instances

The scenario case is handled using the proposed HACA-ST algorithm; the resulting solution is compared with the exact solution of the CPLEX solver. The running time of the CPLEX solver is set to the largest value of 7200 s (in fact, the optimal solution can be obtained within 4500 s for both scenario instances, as shown in Table 6), where the CPLEX 12.5 solver is implemented using the generalized algebraic modeling system (GAMS) platform. HACA-ST parameters are the same as above, and the HACA-ST algorithm is run independently for 10 times. The comparison results are shown in Table 6. From the results it can be seen that the optimal values of the HACA-ST algorithm can reach the level of the exact solution of the solver. The GAP between the average value and the exact solution is <3%, but the solving time is much less than that of the CPLEX solver. Therefore, in the case test of this scenario, the algorithm in this study obtains high-quality solutions with shorter computation time and is more practical than the CPLEX solver.

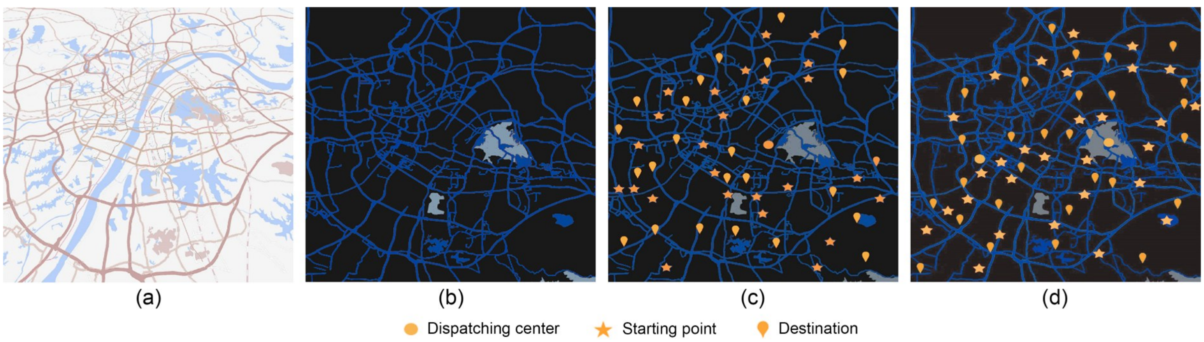

Fig. 10 User distribution of scenario instances: (a) a region of Wuhan, China; (b) road network; (c) user distribution of case 1; (d) user distribution of case 2

Table 6 Comparison results of the scenario instances

Case	HACA-ST			CPLEX		
	Min	Avg	Time (s)	UB	LB	Time (s)
1	888.26	905.36	187	888.26	888.26	3807
2	928.67	946.85	202	928.67	928.67	4261

Min: minimum; Avg: average; UB: upper bound; LB: lower bond

The optimal route according to the HACA-ST algorithm is shown in Fig. 11. Fig. 11a shows the optimal route for case 1, and Fig. 11b shows the optimal route for case 2. Both the horizontal and vertical axes are in kilometers. The vehicles depart from the dispatching center and serve each user in sequence within the time range reserved by each user, eventually completing all user travel and returning to the vehicle dispatching center.

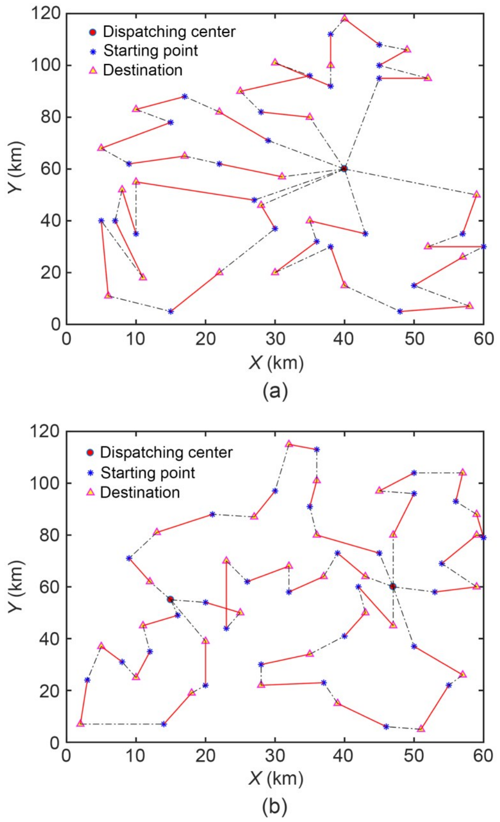

Fig. 11 Optimal routes of the scenario instances: (a) route of case 1; (b) route of case 2

5.5 Performance analysis of HACA-ST

From the simulation results, it can be seen that the HACA-ST algorithm proposed in this study has better results in solving the VRPC problem, and the simulation results of instances are better than those of several other types of optimization algorithms. To further analyze the superiority of the algorithm, different algorithm execution schemes are set up based on the principle of the HACA-ST algorithm, first comparing the effect of solving the algorithm under the spatiotemporal distance and ordinary Euclidean distance, and then comparing the solution concerning whether to introduce spatiotemporal clustering. Different algorithm execution schemes are set up as shown in Table 7.

Table 7 HACA-ST algorithm execution scheme

No.	Distance function	Clustering method
S1	Spatiotemporal distance	Spatiotemporal clustering
S2	Euclidean distance	Spatial clustering
S3	Spatiotemporal distance	No clustering

In this study, S1 includes the complete HACA-ST algorithm proposed, and introduces the spatiotemporal distance function and spatiotemporal clustering. S2 uses the Euclidean distance and introduces the clustering stage. In S3, the spatiotemporal distance is introduced, but there is no spatiotemporal clustering stage. The optimal result obtained each time and the number of iterations to reach the optimal value are recorded. The comparison results are shown in Table 8. Comparison of S1 and S2 shows that the optimal value is better for the result obtained using the spatiotemporal distance function. Comparing S1 and S3, it can be seen that in S1, with the introduction of the clustering stage, the optimal and average values obtained are also significantly smaller than those in S3 obtained without the introduction of clustering, which would make the algorithm more accurate.

Table 8 Comparison results of different schemes

Case	Scheme	Min	Avg	Dev (%)	N_{ave}
1	S1	**888.26**	905.36	1.89	658
	S2	1005.30	1075.30	6.51	702
	S3	935.50	1005.90	7.03	863
2	S1	**928.67**	946.85	1.71	647
	S2	1087.60	1159.50	6.20	739
	S3	988.18	1014.50	8.46	824

Min: minimum; Avg: average; Dev: deviation; N_{ave}: average number of iterations to reach the optimal value (rounded to the nearest integer). The bold font indicates the best result

In summary, the proposed HACA-ST algorithm embeds a spatiotemporal distance function, and introduces a spatiotemporal clustering stage which classifies the problem objects into categories before the heuristic solution. Compared with other algorithms, more heuristic information can be obtained to construct high-quality solutions. Thus, HACA-ST has a greater advantage in dealing with problems with spatiotemporal coupling characteristics.

6 Conclusions

As urban traffic networks become more complex, vehicle scheduling in the car-sharing environment is a critical issue to be solved currently. In this study, a kind of VRPC was developed which considers the operating cost and user experience together. To address the problem, this study first defined a spatiotemporal distance function based on reservation time and position information. Compared with the spatial distance, the spatiotemporal distance could better characterize and quantify the spatiotemporal differences between users. On the basis of the spatiotemporal distance, the HACA-ST algorithm was proposed, and the effectiveness was tested based on test instances and scenario instances.

This study has some shortcomings, however. In the car-sharing service process, it is challenging to predict unexpected events due to different traffic environments. It is difficult for the established VRPC model to fully match realistic scenarios in complex situations. Further research on VRPC modeling under uncertainty could be considered. Also, in terms of VRPC model solving, the proposed HACA-ST algorithm achieved better results by exploiting the spatiotemporal relationship features between users in a static environment. However, when the spatiotemporal properties of the users are ambiguous, the incorrect spatiotemporal relationship features cannot bring positive effects to the solution. How to exploit the spatiotemporal properties more accurately and effectively is the key to improving the applicability of the proposed algorithm, and further improvement of the algorithm could be considered by introducing prediction methods and stochastic programming.

Contributors

Renbin XIAO and Zhenhui FENG designed the research. Zhenhui FENG proposed the approach, performed the experiments, and drafted the paper. Renbin XIAO revised and finalized the paper.

Compliance with ethics guidelines

Zhenhui FENG and Renbin XIAO declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Alemi F, Circella G, Mokhtarian P, et al., 2019. What drives the use of ridehailing in California? Ordered probit models of the usage frequency of Uber and Lyft. *Transp Res Part C Emerg Technol*, 102:233-248. https://doi.org/10.1016/j.trc.2018.12.016

Baños R, Ortega J, Gil C, et al., 2013. A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows. *Comput Ind Eng*, 65(2):286-296. https://doi.org/10.1016/j.cie.2013.01.007

Beheshti AK, Hejazi SR, 2015. A novel hybrid column generation-metaheuristic approach for the vehicle routing problem with general soft time window. *Inform Sci*, 316:598-615. https://doi.org/10.1016/j.ins.2014.11.037

Bonabeau E, Dorigo M, Theraulaz G, 2000. Inspiration for optimization from social insect behaviour. *Nature*, 406(6791): 39-42. https://doi.org/10.1038/35017500

Brandão J, 2020. A memory-based iterated local search algorithm for the multi-depot open vehicle routing problem. *Eur J Oper Res*, 284(2):559-571. https://doi.org/10.1016/j.ejor.2020.01.008

Chapman DA, Eyckmans J, van Acker K, 2020. Does car-sharing reduce car-use? An impact evaluation of car-sharing in Flanders, Belgium. *Sustainability*, 12(19):8155. https://doi.org/10.3390/su12198155

Chen DL, Yao MD, Liu H, 2020. Research on Optimization Method and Platform of Car-Sharing Scheduling. Publishing House of Electronics Industry, Beijing, China, p.76-86 (in Chinese).

Chen YY, Wu HS, Xiao RB, 2022. Improved wolf pack algorithm for UAV path planning problem. *Int J Swarm Intell Res*, 13(1):1-22. https://doi.org/10.4018/IJSIR.302605

Dang YB, Allen TT, Singh M, 2022. A heterogeneous vehicle routing problem with common carriers and time regulations: mathematical formulation and a two-color ant colony search. *Comput Ind Eng*, 168:108036. https://doi.org/10.1016/j.cie.2022.108036

Dantzig GB, Ramser JH, 1959. The truck dispatching problem. *Manag Sci*, 6(1):80-91. https://doi.org/10.1287/mnsc.6.1.80

Dorigo M, Gambardella LM, 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans Evol Comput*, 1(1):53-66. https://doi.org/10.1109/4235.585892

Elsedimy E, Algarni F, 2022. MOTS-ACO: an improved ant colony optimiser for multi-objective task scheduling optimisation problem in cloud data centres. *IET Netw*, 11(2): 43-57. https://doi.org/10.1049/ntw2.12033

Errico F, Desaulniers G, Gendreau M, et al., 2018. The vehicle routing problem with hard time windows and stochastic service times. *Euro J Transp Log*, 7(3):223-251. https://doi.org/10.1007/s13676-016-0101-4

Feng ZH, Xiao RB, 2022. Extended ant colony algorithm based on mixed feedback mechanism. *Contr Dec*, 37(12):3160-3170 (in Chinese). https://doi.org/10.13195/j.kzyjc.2021.0846

Hu SH, Chen P, Lin HF, et al., 2018. Promoting carsharing attractiveness and efficiency: an exploratory analysis. *Transp Res Part D Transp Environ*, 65:229-243. https://doi.org/10.1016/j.trd.2018.08.015

Jia YH, Mei Y, Zhang MJ, 2022. A bilevel ant colony optimization algorithm for capacitated electric vehicle routing problem. *IEEE Trans Cybern*, 52(10):10855-10868. https://doi.org/10.1109/TCYB.2021.3069942

Jiang HW, Guo T, Yang Z, 2022. Research progress of vehicle routing problem. *Acta Electron Sin*, 50(2): 480-492 (in Chinese). https://doi.org/10.12263/DZXB.20201154

Jiang HZ, Zhang XY, 2019. An experimental model of regulating the sharing economy in China: the case of online car hailing. *Comput Law Secur Rev*, 35(2):145-156. https://doi.org/10.1016/j.clsr.2018.12.008

Li G, Wang GG, Xiao RB, 2022. A novel adaptive weight algorithm based on decomposition and two-part update strategy for many-objective optimization. *Inform Sci*, 615:323-347. https://doi.org/10.1016/j.ins.2022.09.057

Lin S, Kernighan BW, 1973. An effective heuristic algorithm for the traveling-salesman problem. *Oper Res*, 21(2):498-516. https://doi.org/10.1287/opre.21.2.498

Ma J, Chen LX, Mahmood A, 2019. One-way car-sharing system based on recharging strategy. Chinese Control Conf, p.2290-2294. https://doi.org/10.23919/ChiCC.2019.8865185

Meng XP, Pian ZY, Shen ZY, 2013. Ant algorithm based on direction-coordinating. *Contr Dec*, 28(5):782-786 (in Chinese). https://doi.org/10.13195/j.cd.2013.05.145.mengxp.017

Nourinejad M, Roorda MJ, 2014. A dynamic carsharing decision support system. *Transp Res Part E Log Trans Rev*, 66: 36-50. https://doi.org/10.1016/j.tre.2014.03.003

Qi MY, Lin WH, Li N, et al., 2012. A spatiotemporal partitioning approach for large-scale vehicle routing problems with time windows. *Trans Res Part E Log Trans Rev*, 48(1):248-257. https://doi.org/10.1016/j.tre.2011.07.001

Sitek P, Wikarek J, Rutczyńska-Wdowiak K, et al., 2021. Optimization of capacitated vehicle routing problem with alternative delivery, pick-up and time windows: a modified hybrid approach. *Neurocomputing*, 423:670-678. https://doi.org/10.1016/j.neucom.2020.02.126

Solomon MM, 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper Res*, 35(2):254-265. https://doi.org/10.1287/opre.35.2.254

Sowmya R, Sankaranarayanan V, 2022. Optimal vehicle-to-grid and grid-to-vehicle scheduling strategy with uncertainty management using improved marine predator algorithm. *Comput Electr Eng*, 100:107949. https://doi.org/10.1016/j.compeleceng.2022.107949

Wang WJ, 2010. Improved genetic algorithm for vehicle routing problem with time windows. Int Conf on Intelligent Computing and Cognitive Informatics, p.203-206. https://doi.org/10.1109/ICICCI.2010.42

Wang Y, Zhang J, Assogba K, et al., 2018a. Collaboration and transportation resource sharing in multiple centers vehicle routing optimization with delivery and pickup. *Knowl-Based Syst*, 160:296-310. https://doi.org/10.1016/j.knosys.2018.07.024

Wang Y, Assogba K, Liu Y, et al., 2018b. Two-echelon location-routing optimization with time windows based on customer clustering. *Expert Syst Appl*, 104:244-260. https://doi.org/10.1016/j.eswa.2018.03.018

Wang Y, Wang L, Chen GC, et al., 2020. An improved ant colony optimization algorithm to the periodic vehicle routing problem with time window and service choice. *Swarm Evol Comput*, 55:100675. https://doi.org/10.1016/j.swevo.2020.100675

Wang Y, Ran LY, Guan XY, et al., 2022. Collaborative multicenter vehicle routing problem with time windows and mixed deliveries and pickups. *Expert Syst Appl*, 197:116690. https://doi.org/10.1016/j.eswa.2022.116690

Xiao RB, Chen ZZ, 2023. From swarm intelligence optimization to swarm intelligence evolution. *J Nanchang Inst Technol*, 42(1):1-10 (in Chinese).

Xiao RB, Wang YC, 2018. Labour division in swarm intelligence for allocation problems: a survey. *Int J Bio-Inspir Comput*, 12(2):71-86. https://doi.org/10.1504/IJBIC.2018.094186

Xiao RB, Feng ZH, Wang JH, 2022. Collective intelligence: conception, research progresses and application analyses. *J Nanchang Inst Technol*, 41(1):1-21 (in Chinese). https://doi.org/10.3969/j.issn.1006-4869.2022.01.002

Yang XT, Zhang T, Bai LP, et al., 2019. Appointment scheduling and routing problem of community-home-health-care: based on modified ant-colony algorithm. *Syst Eng Theory Pract*, 39(5):1212-1224 (in Chinese). https://doi.org/10.12011/1000-6788-2017-1328-13

Yu VF, Jodiawan P, Redi AANP, 2022. Crowd-shipping problem with time windows, transshipment nodes, and delivery options. *Transp Res Part E Log Transp Rev*, 157:102545. https://doi.org/10.1016/j.tre.2021.102545

Zhang JL, Zhao YW, Wang HW, et al., 2017. Multi-objective cooperative QEA for low-carbon time dependent vehicle routing problem with simultaneous delivery and pickup. *Int J Wirel Mob Comput*, 12(4):400. https://doi.org/10.1504/IJWMC.2017.085567

Zhang WQ, Yang DJ, Zhang GH, et al., 2020. Hybrid multiobjective evolutionary algorithm with fast sampling strategy-based global search and route sequence difference-based local search for VRPTW. *Expert Syst Appl*, 145:113151. https://doi.org/10.1016/j.eswa.2019.113151

Zhang WY, Xia DW, Chang GY, et al., 2022. APFD: an effective approach to taxi route recommendation with mobile trajectory big data. *Front Inform Technol Electron Eng*, 23(10): 1494-1510. https://doi.org/10.1631/FITEE.2100530

List of supplementary materials

1 Supplement to the VRPC model
2 Supplement to the HACA-ST algorithm
3 Supplement to numerical examples