Applied Soft Computing 139 (2023) 110203

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

An ant colony optimization based on local search for the vehicle routing problem with simultaneous pickup–delivery and time window

Hongguang Wu [a], Yuelin Gao [b,c,*]

[a] School of Mathematics and Statistics, Ningxia University, Yinchuan, 750021, China
[b] Ningxia Province Cooperative Innovation Center of Scientific Computing and Intelligent Information Processing, North Minzu University, Yinchuan, 750021, China
[c] Ningxia Province Key Laboratory of Intelligent Information and Data Processing, North Minzu University, Yinchuan, 750021, China

ARTICLE INFO

Article history:
Received 3 October 2022
Received in revised form 12 February 2023
Accepted 6 March 2023
Available online 27 March 2023

Dataset link: https://neo.lcc.uma.es/vrp/vrp-instances/, https://www.bernabe.dorronsoro.es/vrp

MSC:
00-01
99-00

Keywords:
Vehicle routing problem
Ant colony optimization
Random transition rule with direction
Destory operator
Repair operator

ABSTRACT

The Vehicle Routing Problem with Simultaneous Pickup–Delivery and Time Window (VRPSPDTW) is an important logistics distribution problem. Due to the complexity of this problem, there are few researches on it and lack of relevant solutions. To solve this problem, this paper proposes to use the ant colony optimization (ACO) for the first time, which a swarm intelligence optimization algorithm. An ant colony optimization algorithm with destory and repair strategies (ACO–DR) is proposed on the basis of ACO. Firstly, ACO–DR designs a random transition rule with direction to improve the probability of the algorithm to search the target and to enhance the global search ability of the algorithm. Secondly, because the positive feedback property of ACO, it is easy for the algorithm to fall into the local optimum. Therefore, two local operators, the destory operator and the repair operator, are added to avoid this phenomenon. Finally, to verify the performance of the proposed ACO-DR algorithm, it is tested on Solomon benchmark and Gehring–Homberge benchmark and compared with the state-of-the-art algorithms. The experimental results show that the ACO-DR algorithm is feasible and provides a new effective algorithm for solving VRPSPDTW problem. Besides, the proposed algorithm also has practical implications for vehicle routing problem and the results show that it is applicable and effective in practical problems.

1. Introduction

Nowadays, the development of logistics industry gradually tends to be globalized and informationized, and the role of logistics distribution in the whole logistics system becomes more and more important. In the process of logistics distribution, vehicle transportation is the most important link. Whether the transportation route of goods is reasonable or not directly affects the cost and benefit of distribution, so it is necessary to select the proper vehicle route solution to improve the service quality and enhance the customer satisfaction of the logistics link.

In the process of logistics distribution, the goods in the distribution center need to be transported to different destinations by truck groups. With the development and progress of the society, distribution centers can not only provide delivery services for customers, but also provide pickup services to customers. Therefore, customers can have both delivery demand and pickup demand at the same time. In the vehicle routing problem (VRP), the problem in which customers have both delivery and pickup demands is called vehicle routing problem with simultaneous pickup and delivery (VRPSPD). The VRPSPD problem was proposed by Min [1] in 1989.

VRPSPD problem is the basis of this paper to study the vehicle routing problem with simultaneous pickup-delivery and time window (VRPSPDTW). The VRPSPDTW problem refers to providing both forward supply service and reverse recovery service to customers in a certain period of time, and how to dispatch a fleets at distribution centers to meet customer needs with the least amount of vehicles and travel costs.

At present, there are two main aspects of research on VRPSPDTW, one is VRPSPDTW, the other is VRPSPDTW in home health care. Firstly, for the study of VRPSPDTW problem, Lai et al. [2] established a mixed integer programming model for VRPSPDTW and proposed an improved differential evolution (DE) algorithm to solve it. The algorithm uses a new decimal code

* Corresponding author at: Ningxia Province Cooperative Innovation Center of Scientific Computing and Intelligent Information Processing, North Minzu University, Yinchuan, 750021, China.
E-mail addresses: 2440576496@qq.com (H. Wu), gaoyuelin@263.net (Y. Gao).

https://doi.org/10.1016/j.asoc.2023.110203
1568-4946/© 2023 Published by Elsevier B.V.

to construct the initial population, introduces a penalty technique to publish infeasible solutions, and designs an adaptive crossover probability that varies with the number of iterations in the crossover operation. The numerical results show that this method is effective for solving VRPSPDTW. Wang et al. [3] established a hybrid binary integer programming model with minimizing the number of vehicles (NV) as the main objective for VRPSPDTW, and proposed a co-evolutionary genetic algorithm (GA) with the cheapest insertion method to speed up the solution. Wang et al. [4] established a mixed integer programming model focusing on minimizing the number of vehicles when solving VRPSPDTW, and proposed a simulated annealing (SA) algorithm to solve this NP-hard optimization problem. Wang et al. [5] proposed a heuristic parallel simulated annealing (P-SA) algorithm based on residual capacity and radial surcharge (RCRS) insertion to minimize vehicle path cost from both vehicle cost and vehicle running cost, and applied it to solve VRPSPDTW. Hof et al. [6] developed a hybrid heuristic solution method for VRPSPDTW, which combined the adaptive large neighborhood search algorithm (ALNS) with the path reconnection (PR) method, called ALNS-PR, and proved that the proposed algorithm is competitive by testing it on benchmark instances. Shi et al. [7] proposed an effective learning-based two-stage algorithm to solve VRPSPDTW. In the first stage, a modified variable neighborhood search method with a learning-based objective function is proposed to minimize the main objective with retaining the potential structure. In the second stage, a bi-structure based tabu search (BSTS) is designed to further optimize the primary and secondary target further. Liu et al. [8] proposed a meme algorithm (MATE) with efficient local search and extended neighborhood to solve the VRPSPDTW problem. The MATE algorithm's initialization process, crossover, and large step operations enable it to explore the search space more efficiently.

At the same time, studies on variants of VRPSPDTW have gradually emerged. Wang et al. [9] defined a general multi-objective VRPSDPTW (MO-VRPSDPTW) with five objectives, designed and implemented the multi-objective local search (MOLS) and multi-objective meme algorithm (MOMA) for solving MO-VRPSDPTW. Li et al. [10] studied the discrete multi-objective VRPSDPTW using the decomposition based multi-objective chemical reaction optimization method. To balance diversity and convergence, the authors designed invalidation collision operators on walls and invalidation collision operators between molecules for local search, as well as decomposition and synthesis operators to enhance global convergence. Wang et al. [11] studied the cooperative multi-center vehicle routing problem with time window and hybrid delivery and pickup (CMVRPTWMDP). A mixed integer programming model is established to minimize the logistics operation cost by considering the impact of traffic resource sharing on reducing the number of vehicle demands and maintenance costs. A two-stage hybrid algorithm combining customer clustering and vehicle routing optimization is designed to solve CMVRPTWMDP.

Secondly, the research on VRPSPDTW in home health care (VRPSPDTW-HHC). Liu et al. [12] first introduced the vehicle scheduling problem in the home health care industry. To solve this problem, two meta-heuristic algorithms, hybrid genetic algorithm and tabu search algorithm, are proposed. The authors tested both algorithms using a set of instances generated from a classical VRPTW benchmark. The results show that these two methods are better than the construction heuristic. Liu et al. [13] proposed genetic algorithm (GA) and tabu search (TS) methods to solve VRPSPDTW-HHC. Chaieb et al. [14] proposed a hierarchical method to solve the home health care scheduling problem with simultaneous pickup and delivery and time Windows (HHSP-SPDTW). In Table 1, we summarize and list the relevant literature on VRPSPDTW.

However, the VRPSPDTW problem is closely related to practice and has complexity, and the problem itself is a NP-hard problem, so there are relatively few researches and literatures on this aspect. The VRPSPDTW problem is mainly to reduce the total cost of logistics enterprises by reducing the number of vehicles and the total distance. In the existing literature, the primary objective is to reduce the number of vehicles (NV), and the secondary objective is to reduce the total distance (TD). According to the results of existing studies, it is found that optimizing NV does not necessarily reduce TD, and optimizing TD does not necessarily reduce NV, there are inconsistencies in the direction of NV and TD reduction. This phenomenon occurs because in the actual situation, reducing the enterprise cost includes reducing the fixed cost of vehicle use and reducing the path cost of vehicle travel route. However, the reduction of vehicle routes will not only involve the reduction of enterprise costs, but also affect the efficiency of goods distribution, which will lead to the problem of cost and efficiency. Since cost and efficiency are an efficiency backward relationship, thus making NV and TD also have this relationship to some extent. To solve the above phenomenon, this paper considers the combination with the actual situation when studying the VRPSDPTW problem, with the aim of weighing the relationship between the cost of the enterprise and the transportation efficiency to maximize the profit of the enterprise. The cost of the enterprise is related to the number of vehicles and the transportation efficiency is related to the transportation distance.

The objective function of the VRPSPDTW problem in the existing literature as a manifestation of the weighted function for optimizing the transport distance and the number of vehicles, which has a strong flexibility because the decision maker can choose personal preference according to changing the weight value, i.e., giving priority to optimizing the transport distance or giving priority to optimizing the number of vehicles. Therefore, based on this point, this paper still adopts the method of weighting function when establishing the objective function, but changes the mathematical formula of the weights. Different from the existing literature, the study of VRPSPDTW in this paper will focus on the practical connection, and the weights used in this paper make the objective function not only enable decision makers to choose the optimization objective flexibly, but also help enterprises to weigh the relationship between cost and efficiency when making logistics decisions so that the economic benefits of enterprises can be maximized. Since there is a backward relationship between cost and efficiency, considering the impact of cost on efficiency, it is not better to have lower cost, too low cost may affect the efficiency of the enterprise and lead to the loss of enterprise revenue. Therefore, it is feasible to reduce costs appropriately to improve efficiency and maximize economic benefits.

In addition, according to the summary of VRPSDPTW literature in Table 1, it can be seen that most of the methods for solving VRPSDPTW problems are meta-heuristic algorithms, such as DE, GA, SA, TS, etc. At present, no researchers have used ant colony optimization algorithm (ACO) to solve this problem. In this paper, ACO is proposed to solve VRPSPDTW problem for the first time. The purpose is to develop and explore an effective swarm intelligence algorithm to solve VRPSDPTW problem. The reason why ACO algorithm is considered to solve VRPSDPTW problem is that ACO is a powerful tool to solve discrete optimization problems, and VRPSPDTW belongs to the category of discrete optimization.

ACO proposed by Dorigo et al. [15] is a swarm intelligent optimization algorithm, which has a wide range of applications in real life. Among them, the most widespread and important application of ACO is in path planning, such as fire evacuation route planning [16], ship pipeline path design [17], robot system path planning [18], power routing optimization [19], traveling

Table 1
Literature review of the VRPSPDTW problem.

		Author	Year	Solution	Benchmark
VRPSPDTW		Lai and Chao [2]	2010	Differential Evolution (DE)	Practical problem
		Wang and Chen [3]	2012	Genetic Algorithm (GA)	Solomon
		Wang et al. [4]	2013	Simulated Annealing (SA)	Solomon
		Wang et al. [5]	2015	Simulated Annealing (SA)	Solomon+Gehring-Homberger
		Hof and Schneider [6]	2019	Adaptive Large Neighborhood Search (ALNS)	Solomon+Gehring-Homberger
		Shi et al. [7]	2020	Lexicographic-based two-stage Algorithm	Solomon+Gehring-Homberger
		Liu et al. [8]	2021	Memetic Search	Solomon+Jdset
VRPSPDTW' variants	MO-VRPSPDTW	Wang et al. [9]	2016	Multi-objective Local Search (MOLS)	Real-word MO-VRPSPDTW instances
		Li et al. [10]	2018	Chemical Reaction Optimization (CRO)	Real-word MO-VRPSPDTW instances
	CMVRPTWMPD	Wang et al. [11]	2022	GA+Oartical Swarm Optimization (PSO)	Solomon
VRPSPDTW in home health care	VRPSPDTW-HHC	Liu et al. [12]	2012	GA+Tabu Search (TS)	Solomon
		Liu et al. [13]	2013	GA+Tabu Search (TS)	Solomon+Gehring-Homberger
	HCSPSPDTW	Chaieh and Sassi [14]	2021	Tabu Search (TS)	Solomon

salesman problem [20,21], vehicle path planning [22–25], etc. In the planning of VRP, ACO can be used to solve VRP with time window [26–28] and VRPSPD [29–31], but for the VRPSPDTW problems studied so far no literature has appeared on solving with ACO. When picking up and delivering, it usually gives priority to customers with short time and urgent needs, and it also gives priority to customers with short service time. Therefore, it is necessary to take these two factors into account when designing the ACO algorithm to get a solution that better meets the actual needs. The improved ant colony algorithm called ACO-DR that proposed in this paper is based on the basic ACO, proposes a random transition rule with direction, and introduces two operators: destory operator and repair operator to overcome the shortage of ACO. The random selection strategy with direction is proposed to improve the probability of algorithm to search the target, and to enhance the global searching ability of the algorithm. The destory operator and repair operator are introduced to avoid falling into local optimum due to the positive feedback property of the algorithm.

The rest of this paper is organized as follows. The second part introduces the mathematical model of VRPSPDTW problem. The third part is the design of improved ant colony algorithm to solve VRPSPDTW. The fourth part is the numerical experiment analysis. The fifth part is the practical application. The sixth part is the conclusion.

2. Mathematical model of VRPSDPDTW problem

VRPSPDTW can be defined as a directed graph $G = (V, E)$, where $V = \{0, 1, \dots, n\}$ represents the set of all customers, 0 represents the depot, and $V_0 = \{1, 2, \dots, n\}$ represents the customer set, $E = \{(i, j)|i, j \in V_0\}$ represents the edge set, the distance between customer i and customer j is denoted by c_{ij}. In the study of this paper, the specifications of all vehicles $k \in K = \{1, 2, \dots, m\}$ are completely consistent, they have the same maximum speed, maximum running time, loading capacity and empty weight. The vehicle distribution route must start and end at the depot (0). Each customer i has a delivery demand d_i and a pickup demand p_i. d_i represents the quantity of goods delivered from the depot to customer i. p_i represents the quantity of goods pickup from customer i that must be delivered to the depot. It is assumed that different kinds of goods are compatible and can be loaded in the same vehicle. There are no delivery and pickup demand at the depot. The total customer demand for any route cannot exceed the maximum capacity C of the vehicle, where L_{0k} represents the loading capacity of vehicle k leaving the depot, L_j represents the loading capacity of the vehicle after serving customers.

Both the depot and the customer have time windows $[a, b]$, the lower bound a and upper bound b of the time window define the earliest and latest time for the vehicle to leave and return to the depot $[a_0, b_0]$, and the earliest and latest time for the vehicle to serve the customer $[a_i, b_i]$. Each customer can be served by a vehicle only once within a given time window. The time taken for the vehicle to be loaded and unloaded at customer i is denoted by w_i. The time taken by the vehicle to travel from customer i to customer j is denoted by t_{ij}. The starting service time of vehicle k to customer i is denoted by S_{ik}.

According to the above description of the VRPSDPTW problem, the following mathematical model is established:

$$\min \quad \sigma \sum_{j\in V}\sum_{k\in K} g_d x_{0jk} + (1-\sigma)\sum_{i\in V}\sum_{j\in V}\sum_{k\in K} g_t c_{ij} x_{ijk} \tag{1}$$

$$s.t. \quad \sum_{i\in V}\sum_{k\in K} x_{ijk} = 1, \forall j \in V_0 \tag{2}$$

$$\sum_{i\in V} x_{ihk} = \sum_{j\in V} x_{hjk}, \forall h \in V_0; \forall k \in K \tag{3}$$

$$\sum_{j\in V} x_{0jk} = 1, \forall k \in K \tag{4}$$

$$\sum_{j\in V_0} x_{0jk} = \sum_{i\in V_0} x_{i0k}, \forall k \in K \tag{5}$$

$$\sum_{i\in V} x_{i0k} = 1, \forall k \in K \tag{6}$$

$$S_{ik} + w_i + t_{ij} - M(1 - \sum_{k\in K} x_{ijk}) \le S_{jk}, \forall i \in V; \forall j \in V \tag{7}$$

$$a_i \le S_{ik} \le b_i, \forall i \in V; \forall j \in V; \forall k \in K \tag{8}$$

$$L_{0k} = \sum_{i\in V}\sum_{j\in V_0} d_j x_{ijk}, \forall k \in K \tag{9}$$

$$L_j \ge L_{0k} - d_j + p_j - M(1 - x_{0jk}), \forall j \in V_0; \forall k \in K \tag{10}$$

$$L_j \ge L_{0k} - d_j + p_j - M(1 - \sum_{k\in K} x_{ijk}), \forall i \in V_0; \forall j \in V_0; i \ne j \tag{11}$$

$$L_{0k} \le C, \forall k \in K \tag{12}$$

$$L_j \le C + M(1 - \sum_{i\in V} x_{ijk}), \forall j \in V; \forall k \in K \tag{13}$$

$$x_{ijk} \in \{0, 1\}, \forall i \in V_0; \forall j \in V_0; \forall k \in K \tag{14}$$

In the above model, σ represents the parameter that trade-offs between scheduling cost and travel cost. g_d represents the scheduling cost of vehicle k. g_t represents the cost of travel or transportation per unit of distance. M stands for a sufficiently large positive number. The 0–1 variable x_{ijk} indicates whether vehicle k starts from customer i and goes to customer j; if so, $x_{ijk} = 1$; otherwise, $x_{ijk} = 0$.

The objective function (1) is to minimize the cost. Constraint (2) restricts each customer to be assigned to only one path. Constraint (3) ensures that each customer is served by the same vehicle. Constraints (4)–(6) ensure that each vehicle starts from the depot, arrives at the customer and must leave for another customer, finally returns to the depot. Constraints (7)–(8) represent the constraints of the time window to ensure the feasibility of the time plan. Constraint (9) is the vehicle load constraints associated with initial vehicle loads. Constraint (10) is the vehicle loads after the first customer. Constraint (11) is the vehicle load constraint associated with the "en route" vehicle load. Constraints (12) and (13) are vehicle capacity constraints, which require that the vehicle specified capacity is never exceeded by the quantity of goods. Constraints (14) specify the variables used in the formulation.

In this study, the primary objective is to reduce the NV and the secondary objective is to reduce the TD. Due to the complexity of the actual situation, it is feasible to reduce the NV to reduce the cost or shorten the TD to improve the distribution efficiency. To facilitate our later discussion of the merits of the solutions obtained by the algorithm, here we define the order of priority of the evaluated solutions. Let S1 and S2 be two solution. S1 gets the value of NV is NV_{S1}, the value of TD is TD_{S1}; S2 gets the value of NV is NV_{S2}, the value of TD is TD_{S2}.

(1) If NV and TD are falling in the same direction:
$NV_{S1} < NV_{S2}$ and $TD_{S1} < TD_{S2}$, then S1 dominates S2;
$NV_{S1} > NV_{S2}$ and $TD_{S1} > TD_{S2}$, then S2 dominates S1;
(2) If NV and TD are falling in the different direction:
$NV_{S1} > NV_{S2}$ and $TD_{S1} < TD_{S2}$;
$NV_{S1} < NV_{S2}$ and $TD_{S1} > TD_{S2}$;
In this case, the solution is determined according to the objective function.
(3) If $NV_{S1} = NV_{S2}$, $TD_{S1} < TD_{S2}$, then S1 dominates S2;
$TD_{S1} = TD_{S2}$, then S1 is equals S2;
$TD_{S1} > TD_{S2}$, then S2 dominates S1.

3. The improved ant colony optimization algorithm is proposed to solve VRPSPDTW

3.1. The general framework

Here, we refer to the proposed improved ant colony optimization algorithm as ACO-DR. The pseudocode of ACO-DR is shown in Algorithm 1. As can be seen from Algorithm 1, at first, ACO-DR needs to receive relevant data of the problem, such as the location of nodes, pickup demand, delivery demand, time window and other values. Then begin the initialization steps, set the initial value of iteration: iter $=$ 1, maximum number of iterations: itermax $=$ 200 and set the parameters of ACO. The next step is the iterative search for the best path, which is the core of the whole algorithm. First of all, each ant visits each node until all nodes are visited, and the solution space $S = (S^1, \ldots, S^m)$ of the problem is constructed, where $S^i (i = 1, \ldots, m)$ is the path obtained by each ant visiting the node according to the random transfer rule with direction. Then, the length of each ant's path is calculated, and the best path S_{best} in the current iteration is recorded. Because ACO has the characteristic of positive feedback, it is easy to make the algorithm fall into local optimum. To avoid this phenomenon, we introduce the destory operator and repair operator. So continue to perform destory and repair operations on S_{best}. The solution S_d is obtained after using the destory operation on S_{best}, and the solution S_r is obtained after using the repair operation on S_d. If the solution S_r is better than S_{best}, then use S_r replacements S_{best}. After the above steps are completed, update the pheromone on the connection path between each node, and finally the optimal path is found after several iterations. Once again using the local search operator for the optimal solution. In the following sections, the random transition rule with orientation, the destory operator, the repair operator, and the global update rule are discussed in detail.

Algorithm 1 Pseudo-code of the Proposed Algorithm ACO-DR

```
1: Input the relevant data of the problem; // input the location of the node, pickup demand, delivery demand, time window, and other values
2: /*Initialization*/
3: Set the initial value of the iterations iter=1 and the maximum number of iterations itermax=200; set the parameters of ACO;
4: /*Main loop: iterates to find the best route*/
5: while $iter \leq itermax$ do
6:     for i=1 to m do // m is the number of ants
7:         for j=1 to n do // n is the number of customer
8:             By using the random transfer rule with direction, select the next node and place the nodes that meet the constraints into the path record table; // Section 3.2
9:         end for
10:    end for
11:    Calculate the length of each ant's path and the shortest path length,record the best path $S_{best}$ in the current iteration number;
12:    Apply the destroy operator to $S_{best}$ to get the solution $S_d$; // Section 3.3.1
13:    Apply the repair operator to $S_d$ to get the solution $S_r$; // Section 3.3.2
14:    if $S_r < S_{best}$ then
15:        $S_{best}$=$S_r$
16:    end if
17:    Update the global pheromone; // Section 3.4
18:    iter=iter+1;
19:    Clear the path record table;
20: end while
21: Once again using the local search operator for the optimal solution
22: Output $S_{best}$
```

3.2. Random transfer rule with direction

To construct a solution, ant k needs to visit all the nodes and needs to judge the movement probability of the alternative nodes. When judging optional nodes, ants not only consider pheromone concentration and heuristic function, but also consider customer time urgency and the service customer time duration. Therefore, time width and customer service time are added to the basic ACO state transition rules to meet the actual situation. Therefore, when ant k moves from node i to node j using the random transition rule with direction in Eq. (15):

$$j = \begin{cases} max_{j \in allowed_k}\{\tau_{ij}^{\alpha}(t) \cdot \eta_{ij}^{\beta}(t) \cdot (\frac{1}{width_j})^{\gamma} \cdot (\frac{1}{wait_j})^{\delta}\}, & r \leq r_0 \\ P_{ij} \qquad , & r > r_0 \end{cases} \tag{15}$$

where $\tau_{ij}^{\alpha}(t)$ represents the amount of pheromone associated with node j, and expresses the subsequent effect from node i to j (indicating the superiority of performing such a move in the previous route). $\eta_{ij}(t) = \frac{1}{d_{ij}}$ is the heuristic function, which represents the expected degree of ant transfer from node i to node j. $allowed_k$ is the set of nodes to be visited by ant k. At the beginning, $allowed_k$ includes all nodes except the starting node of ant k, as time goes on, the number of nodes in $allowed_k$ is gradually reduced until it is empty, indicating that all nodes are accessed. α is the importance factor of pheromone. The greater the value, the greater the role of pheromone concentration in the process of transfer. β is the importance factor of the heuristic function, and the greater the value, the greater the role of the heuristic function in the process of transfer. $width_j = b_j - a_j$ represents the width of the time window, γ represents the importance factor of time span. $wait_j$ represents the duration of service nodes, δ represents the importance factor of waiting time.

Where r is a random number between 0 and 1, $r_0 \in [0, 1]$ is a user-defined value. The random number r is introduced to improve the global search ability of the algorithm and give full play to its exploration ability. If $r \leq r_0$, then the node selected by ant k in the next step is the node with the highest amount of pheromone, the highest heuristic information, the smaller time window and the shorter service time. Otherwise, the ant selects the next node according to the probability formula P_{ij}, it is expressed in Eq. (16):

$$P_{ij} = \frac{\tau_{ij}^{\alpha}(t) \cdot \eta_{ij}^{\beta}(t) \cdot (\frac{1}{width_j})^{\gamma} \cdot (\frac{1}{wait_j})^{\delta}}{\sum_{j \in allowed_k} \tau_{ij}^{\alpha}(t) \cdot \eta_{ij}^{\beta}(t) \cdot (\frac{1}{width_j})^{\gamma} \cdot (\frac{1}{wait_j})^{\delta}} \tag{16}$$

3.3. Destory operator and repair operator

The destory operator is to remove customers from the current solution, and the repair operator is to insert the removed customers back into the destory solution. The process of removing and reinserting is the process of searching the neighborhood, so as to avoid the algorithm falling into local optimum. In this process, the selection of the removed and reinserted customers. Here, we use a common selection strategy: "relevance selection". The concept of relevance is introduced to retain customers with good relevance and remove customers with poor relevance during reinsertion. Generally speaking, the smaller the distance between customers geographically, the greater the relevance between them, otherwise, the opposite is true. In other words, neighboring customers are more correlated than distant customers. In the VRP problem, if two customers are served by the same vehicle, that is, two customers on the same route are also considered related. When the goal of reducing the number of vehicles is important, it is necessary to remove multiple customers on the same route, since removing all customers on a route is the only way to reduce vehicle use. The details of correlation, destory operator and repair operator are as follows:

3.3.1. Destroy operator

The destroy operator removes relevant customers according to the following correlation equation (17):

$$R_{ij} = \frac{1}{(C'_{ij} + V_{ij})} \tag{17}$$

where C'_{ij} is the normalized value of C_{ij}, which ranges from [0,1]; C_{ij} is the Euclidean distance between customers i and j.

$$V_{ij} = \begin{cases} 0, & customer\ i\ and\ j\ are\ served\ by\ the\ same\ vehicle \\ 1, & otherwise \end{cases}$$

From the above formula (17), it can be seen that the larger R(i,j) is, the greater is the correlation between customers i and j. Based on the above correlation formula, assuming that the number of customers is N, the number of customers to be removed is L, and the random element is D, then the steps of the destroy operator are as follows:

Step1: Randomly select a customer from these N customers, e.g., the randomly selected customer is i, at the point the set of removed customers $\mathbf{R} = [\mathbf{i}]$ and the set of unremoved customers $\mathbf{U} = [\mathbf{remaining\ N - 1\ customers}]$;

Step2: Judge whether the number of customers in set **R** is less than or equal to L. If so, go to Step3. If not, go to Step5;

Step3: First, a customer r is randomly selected from set **R**, and the correlation between all customers in set **U** and customer r is calculated according to the correlation calculation formula. Secondly, the customers in set **U** are ranked according to the order of relevance, and the ranking result is S. Then calculate the next customer selected to be removed according to the formula $\lceil rand^D \rceil \times |U|$ ($rand \in (0, 1)$; $|U|$ is the number of customers in set **U**; $\lceil\ \rceil$ means round up);

Step4: Delete customer next from set **U** and go to Step2;

Step5: Remove all customers in set **R** from the current solution; output the set **R** of removed customers and the destory solution $S_{destory}$.

3.3.2. Repair operator

Complete the destory operation to obtain the removed customer set **R** and the destory solution $S_{destory}$, based on which the repair operator is operated.

There are two important concepts in the repair operator, namely "insertion cost" and "regret value". (1) insertion cost: if the current destory solution is $S_{destory}$, on the premise of not violating the constraints of loading capacity and time window, after inserting a customer in set **R** back into an insertion position in $S_{destory}$, the total distance of the repaired solution minus the total distance is the "insertion cost" of the customer inserted into this position. (2) regret: without violating the constraint, if the total number of feasible insertion positions l_r that customer i in set **R** can be inserted back into $S_{destory}$, then these l_r insertion positions correspond to l_r insertion cost. Next, rank these l_r insertion costs from small to large. If the ranked insertion cost is up_{data}. Then, the regret value of inserting customer i back to $S_{destory}$ is the insertion cost of the second position after sorting minus the insertion cost of the first position, namely $up_{data}(2) - up_{data}(1)$. The specific steps to repair operator are as follows:

Step1: Initialize the repaired solution $S_{destory}$, $S_{repair} = S_{destory}$;

Step2: If set **R** is non-empty, go to Step3, otherwise go to Step6;

Step3: Calculate the current number of customers in set **R** and calculate the "regret value" of inserting each customer in set **R** back to $S_{destory}$;

Step4: Firstly, find the maximum "regret value" corresponding to the serial number max_{index}; secondly, determine the customer $rc = R(max_{index})$ that will be inserted back; finally, insert the rc back into the "insertion cost" minimum position in $S_{destory}$;

Step5: Update $\mathbf{R}(\mathbf{max_{index}}) = [\ ]$, go to Step2;

Step6: End of the repair operation, output the repaired solution S_{repair}.

3.4. Update pheromone

Pheromones are chemicals secreted by ants to communicate with other ants. Ants can be detected by other ants by releasing pheromones in their foraging paths. When more and more ants pass by on a certain path, it means that there will be more and more pheromones on the path. To avoid too many pheromones on the path and causing the pheromone information to overwhelm the inspired information, the pheromones are updated after each ant has taken a step or finished traversing all nodes. This updated strategy is that while new pheromones are added to a certain path, the old pheromones remaining on the path fade out over time until they are absent. Thus, the amount of pheromones on path (i, j) at moment $t + n$ can be updated according to the rules of equations:

$$\tau_{ij}(t + n) = (1 - \rho) \times \tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{18}$$

$$\Delta\tau_{ij}(t) = \sum \Delta\tau_{ij}^{k}(t) \tag{19}$$

$$\Delta\tau_{ij}^{k}(t)=\begin{cases}\frac{Q}{L_s}, & if\ (i,j)\ \in\ S_{best}\\ 0, & otherwise\end{cases} \tag{20}$$

where ρ represents the volatile coefficient of pheromone; $1-\rho$ represents the pheromone residual factor; $\Delta\tau_{ij}(t)$ represents pheromone increment on path (i,j) in this cycle, at the initial time $\Delta\tau_{ij}(t)$ is equal to 0; $\Delta\tau_{ij}^{k}(t)$ represents the amount of information left on path (i,j) by the ant k in this cycle; Q stands for pheromone intensity; L_s denote the length of the path (i,j) in the solution S_{best}.

4. Numerical analysis

In this section, we conduct extensive computational experiments of the proposed ACO-DR algorithm on benchmark instances, and compare the numerical results obtained with the state-of-the-art results in the literature to evaluate the performance of the proposed algorithm. The running environment of the algorithm is MATLAB2016. This section introduces the benchmark instance, then gives the parameters of the running algorithm, and then compares the algorithm and analyzes the results based on the benchmark instance.

4.1. Benchmark instances

The benchmark instance often used in the VRPSPDTW literature is the instance obtained by Wang and Chen [3] with modifications on the Solomon benchmark instance, which is also the test set widely used today in problems with time windows and delivery pickups. The benchmark instance consists of six different problem types: Cdp1, Cdp2, Rdp1, Rdp1, RCdp1, RCdp2. Where type refers to:

Cdp: customer locations are centralized;

Rdp: customer locations are randomly dispersed;

RCdp: customer location is a mix of random and centralized;

Type1: customer's time window is narrow and the vehicle capacity is small;

Type2: customer's time window is wide and the vehicle capacity is large.

4.2. Parameter settings

A key feature of ACO-DR is that the quality of the solution can be controlled by changing the parameter values, that is, the performance of the proposed algorithm is related to parameter tuning. In the field of swarm intelligence and evolutionary computing, there are traditionally two methods for selecting parameter values [32]. The first method is parameter tuning, in which the parameter values are established before the algorithm is run. In this case, the parameter values are fixed in the initialization stage and will not change during the algorithm running. The second method is parameter control, where the parameter values are established during the algorithm run. In this case, the parameter values are given an initial value when starting the algorithm and are changed during the algorithm run.

Since the parameter values are set before the algorithm runs in this paper, they belong to parameter tuning. The method of parameter tuning is to consider the change of each parameter within a predetermined range of change while other parameters remain unchanged. To illustrate how these parameters affect the overall performance of the algorithm, we analyze the optimal solution and key parameters in the randomly selected problem R101.

(1) Parameter m

The relationship between the optimal solution and the parameter m is shown in Fig. 1. When the number of ants is too large,

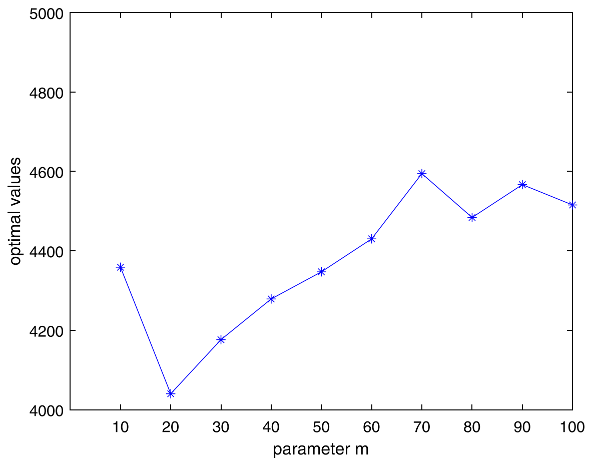

Fig. 1. The relationship between the optimal solution and parameter m.

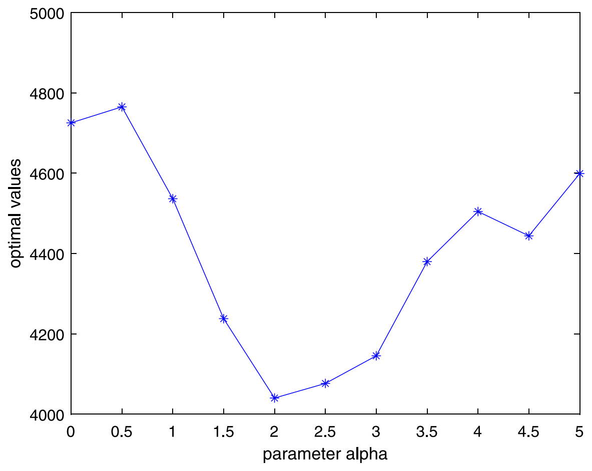

Fig. 2. The relationship between the optimal solution and parameter α.

the changes of pheromones on the path tends to be average, and the randomness is enhanced, but the convergence speed becomes slower. When the number of ants is too small, the randomness decreases and the convergence speed becomes faster, but the stability of the algorithm becomes worse. Therefore, as can be seen from Fig. 1, the performance of our algorithm is relatively better when the value of m is 20.

(2) Parameter α

The relationship between the optimal solution and parameter α is shown in Fig. 2. The value of α has an impact on the randomness of the algorithm, and can lead to the algorithm easily fall into local optimal. Therefore, the performance of our algorithm is relatively better when α is in the interval [2.0, 3.0].

(3) Parameter β

The relationship between the optimal solution and parameter β is shown in Fig. 3. The value of β affects the prior and determinism of the ant colony in the whole optimization process. As can be seen from Fig. 3, the performance of our algorithm is relatively better when β is in the interval [1.0, 1.5].

(4) Parameter γ

The relationship between the optimal solution and parameter γ is shown in Fig. 4. Parameter γ represents the importance factor of time span. As shown in Fig. 4, the performance of our algorithm is relatively better when γ is in the interval [1.5, 2.5].

(5) Parameter δ

The relationship between the optimal solution and parameter δ is shown in Fig. 5. Parameter δ represents the importance factor

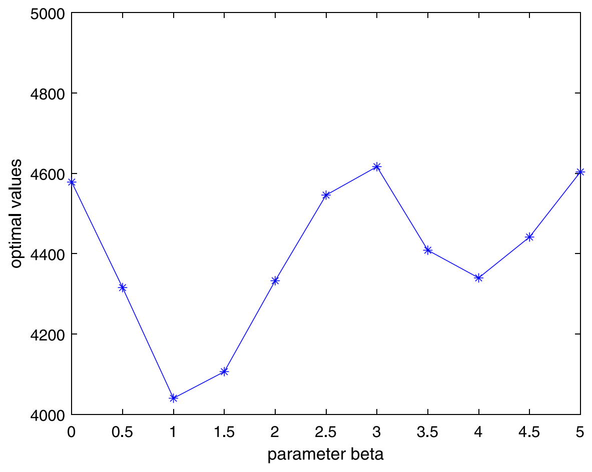

Fig. 3. The relationship between the optimal solution and parameter β.

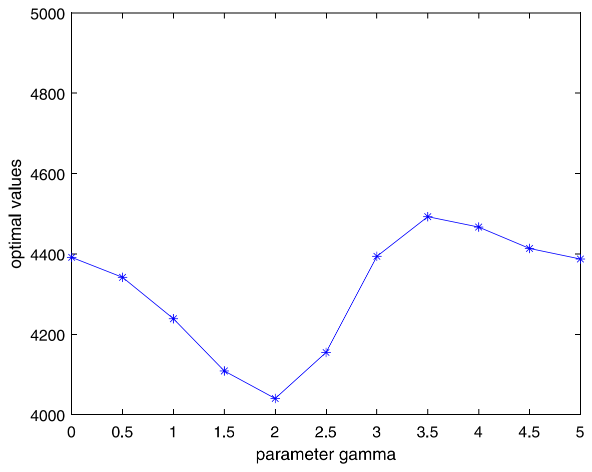

Fig. 4. The relationship between the optimal solution and parameter γ.

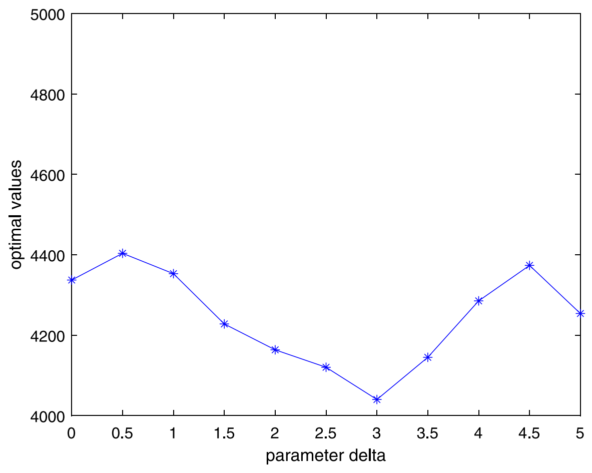

Fig. 5. The relationship between the optimal solution and parameter δ.

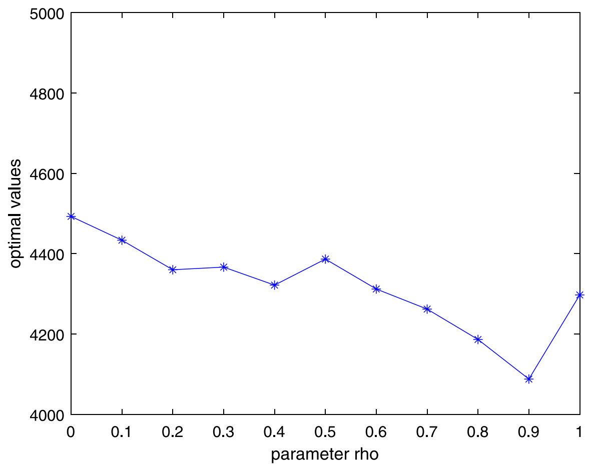

Fig. 6. The relationship between the optimal solution and parameter ρ.

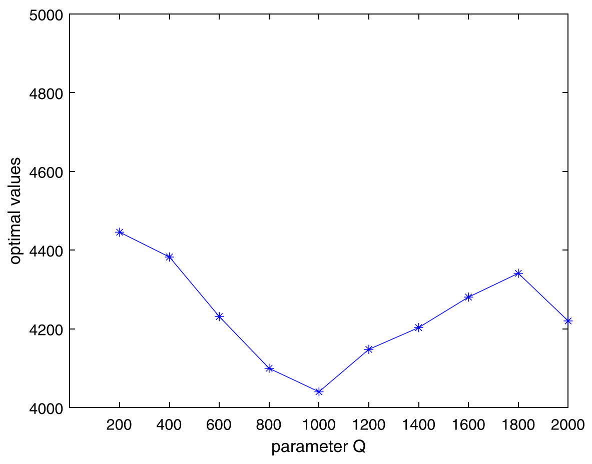

Fig. 7. The relationship between the optimal solution and parameter Q.

of waiting time. As can be seen from Fig. 5, the performance of our algorithm is relatively better when δ is in the interval [2.0, 3.5].

(6) Parameter ρ

The relationship between the optimal solution and parameter ρ is shown in Fig. 6. The value of ρ has an impact on the search ability and convergence speed of the algorithm. The ants will choose their paths at approximately random when the value of ρ is too small. The convergence performance of the algorithm becomes worse when the value of ρ is too large. Therefore, the performance of our algorithm is relatively better when ρ is in the interval [0.8, 0.9].

(7) Parameter Q

The relationship between the optimal solution and parameters Q is shown in Fig. 7. The amount of information Q affects the positive feedback function of the algorithm, so that the algorithm can effectively find the optimal solution of the problem under the effect of positive feedback. Therefore, the performance of our algorithm is relatively better when Q is in the interval [800, 1200].

(8) other parameters

Since different objective functions are used in different literature, the tradeoff parameter σ is used to adjust different decision criteria in this analysis. The primary goal is to minimize the number of vehicles and the secondary goal is to minimize the total distance. Therefore, the value of σ is 0.6, the scheduling cost g_d is 60, the travel cost g_t is 5. In addition, the random number r_0 is a user-defined value, the value of it is 0.5 in this paper.

According to the above analysis of parameter tuning, the parameter values of the algorithm in this paper are determined as: $m = 20$, $\alpha = 2$, $\beta = 1$, $\gamma = 2$, $\delta = 3$, $r_0 = 0.5$, $\alpha = 2$, $\rho = 0.85$, $Q = 1000$, $itermax = 200$.

Table 2
Comparison of experimental results with ACO for Type1 instances.

Instance	Size	ACO		ACO-DR		Gap	
		NV	TD	NV	TD	NV	TD (%)
Rdp101	100	19	1656.26	20	1650.06	+1	−0.37
Rdp102	100	19	1540.03	18	1462.12	−1	−5.06
Rdp103	100	14	1303.38	14	1227.62	0	−5.81
Rdp104	100	11	1138.29	10	1003.58	0	−11.83
Rdp105	100	15	1487.97	14	1383.06	−1	−7.05
Rdp106	100	13	1345.42	12	1281.52	−1	−4.75
Rdp107	100	12	1226.92	11	1092.64	−1	−10.94
Rdp108	100	10	1131.61	10	976.38	0	−13.72
Rdp109	100	13	1313.53	11	1196.88	−2	−8.88
Rdp110	100	12	1266.98	11	1127.55	−1	−11.00
Rdp111	100	12	1232.60	11	1063.06	−1	−13.75
Rdp112	100	11	1167.13	10	1025.84	−1	−12.11
Cdp101	100	12	1099.35	11	970.30	−1	−11.74
Cdp102	100	10	1509.35	10	964.56	0	−36.09
Cdp103	100	10	1403.31	10	913.46	0	−34.91
Cdp104	100	10	1476.04	10	1043.21	0	−29.32
Cdp105	100	11	1083.06	11	989.59	0	−8.63
Cdp106	100	11	1054.78	11	880.01	0	−16.57
Cdp107	100	11	995.28	11	953.14	0	−4.23
Cdp108	100	11	1163.51	10	930.31	−1	−20.04
Cdp109	100	11	1369.56	10	934.43	−1	−31.77
RCdp101	100	17	1732.90	15	1658.58	−2	−4.29
RCdp102	100	14	1535.90	13	1531.25	−1	−0.30
RCdp103	100	12	1546.17	12	1312.07	0	−15.14
RCdp104	100	12	1447.07	11	1183.80	−1	−18.19
RCdp105	100	16	1659.16	14	1569.55	−2	−5.40
RCdp106	100	13	1533.34	13	1420.70	0	−7.35
RCdp107	100	13	1483.90	12	1274.64	−1	−14.10
RCdp108	100	12	1425.92	11	1171.93	−1	−17.81

Table 3
Comparison of experimental results with ACO for Type2 instances.

Instance	Size	ACO		ACO-DR		Gap	
		NV	TD	NV	TD	NV	TD (%)
Rdp201	100	4	1461.80	5	1236.13	1	−15.44
Rdp202	100	4	1430.85	4	1098.90	0	−23.20
Rdp203	100	3	1343.15	3	946.77	0	−29.51
Rdp204	100	3	974.18	3	771.89	0	−20.77
Rdp205	100	3	1331.86	3	1002.98	0	−24.69
Rdp206	100	3	1208.08	3	927.11	0	−23.26
Rdp207	100	3	1078.61	3	852.35	0	−20.98
Rdp208	100	3	1021.99	2	798.14	−1	−21.90
Rdp209	100	3	1167.47	4	894.88	1	−23.35
Rdp210	100	3	1231.53	3	978.89	0	−20.51
Rdp211	100	3	1097.17	3	796.67	0	−27.66
Cdp201	100	3	591.56	3	591.56	0	0
Cdp202	100	4	784.79	3	591.56	−1	−24.62
Cdp203	100	4	977.33	3	591.17	−1	−39.51
Cdp204	100	4	1110.69	3	590.60	−1	−46.83
Cdp205	100	3	622.69	3	588.88	0	−5.42
Cdp206	100	3	697.36	3	588.49	0	−15.61
Cdp207	100	4	725.27	3	588.29	−1	−18.89
Cdp208	100	3	702.55	3	588.32	0	−16.26
RCdp201	100	5	1588.25	4	1383.79	−1	−12.87
RCdp202	100	4	1746.53	4	1197.36	0	−31.44
RCdp203	100	4	1435.63	4	984.59	0	−31.42
RCdp204	100	3	1118.43	3	809.53	0	−27.62
RCdp205	100	4	1568.97	4	1370.54	0	−12.65
RCdp206	100	4	1504.38	3	1137.28	−1	−24.00
RCdp207	100	4	1385.94	4	1032.91	0	−25.47
RCdp208	100	3	1202.19	3	862.66	0	−28.24

4.3. Experimental results and comparison

In the comparison algorithm, we consider five algorithms for solving VRPSPDTW problem, which are CO-GA [3], P-SA [5], ALNS-PR [6], VNS-BSTS [7] and MATE [8]. These five have been tested in whole or in part on benchmark instances, so we compared the best test results of these algorithms directly from the original publication. To compare and analyze the performance of the proposed algorithm comprehensively, our experimental comparison is mainly divided into three parts: (1) ACO-DR is compared to ACO on the Solomon instance; (2) ACO-DR is compared with five state-of-the-art algorithms on Solomon instances; (3) ACO-DR is compared with three state-of-the-art algorithms on Gehring and Homberge instances. All experiments were run independently 20 times. The detailed comparative results and analysis are as follows:

4.3.1. Comparison with ACO on Solomon instances

In this section, we compare the proposed algorithm ACO-DR with the original algorithm ACO on Solomon instances. In this comparison, we mainly discuss the Typ1 and Type2 of Solomon instances. Firstly, for Type1 instances (Rdp1, Cdp1 and RCdp1), the comparison results between ACO-DR and ACO are shown in Table 2.

In Table 2, the first column represents the instance, and the second column represents the size of the instance. For each instance in the table, the optimal NV and TD calculated by ACO and ACO-DR are given. The last column of this table gives the gap between the proposed algorithm ACO-DR and ACO. Where $Gap_{NV} = NV_{ACO-DR} - NV_{ACO}$, $Gap_{TD} = \frac{TD_{ACO-DR} - TD_{ACO}}{TD_{ACO}} \times 100\%$. By observing the data in the gap column, it can be seen that except for the instance Rdp101, when ACO-DR solves such instances as narrow time window and small vehicle capacity, the gap of NV is between [0,−2], while the gap of TD is "−". ACO-DR is superior to ACO in finding solutions, using less NV and shorter TD.

Secondly, for Type2 instances (Rdp2, Cdp2 and RCdp2), the comparison results between ACO-DR and ACO are shown in Table 3.

As can be seen from Table 3, in the instance Rdp208, Cdp202, Cdp203, Cdp204, Cdp207, RCdp201 ACO-DR results are better than ACO, used less NV and TD is shorter. In instances Rdp201 and Rdp209, although ACO-DR gets shorter TD than ACO, it uses more NV than ACO. Therefore, according to the numerical analysis results in Tables 2 and 3, the results obtained by the proposed ACO-DR algorithm are generally better than those obtained by ACO.

4.3.2. Comparison with state-of-the-art algorithms on Solomon instances

In this part, we compare the proposed algorithm with the advanced algorithms for solving VRPSPDTW in the literature, which are CO-GA (2012), P-SA (2015), ALNS-PR (2019), VNS-BSTS (2020) and MATE (2021). The comparison in this section is mainly discussed separately from the customer location distribution of the Solomon instance. Black text indicates that the results are better than or equal to other algorithms. For instances where customer location is randomly generated (Rdp), the results obtained by ACO-DR compared with the advanced algorithm are shown in Table 4.

The analysis of the data in Table 4 shows that in the instances where the customer location is randomly generated, CO-GA algorithm gets 6 better NV values and 4 better TD values than the other algorithms. P-SA algorithm gets 11 better NV values and 2 better TD values than the other algorithms. VNS-BSTS algorithm gets 19 better NV values and 2 better TD values than the other algorithms. ALNS-PR algorithm gets 17 better NV values and 4 better TD values than the other algorithms. MATE algorithm gets 15 better NV values and 8 better TD values than the other algorithms. ACO-DR algorithm gets 4 better NV values and 7 better TD values than the other algorithms. If we compare the overall experimental results of all algorithms, it can be found that the ACO-DR algorithm obtains better results than CO-GA algorithm,

Table 4
Experimental results are compared with advanced algorithms for Rdp instance.

Instance	CO-GA		P-SA		VNS-BTST		ALNS-PR		MATE		ACO-DR	
	NV	TD	NV	TD	NV	TD	NV	TD	NV	TD	NV	TD
Rdp101	19	1653.53	19	1660.98	19	1650.80	19	1650.80	19	1650.80	20	**1650.06**
Rdp102	17	1488.04	17	1491.75	17	1486.12	17	1486.12	17	1486.12	18	**1462.12**
Rdp103	14	**1216.16**	14	1226.77	13	1294.75	13	1597.01	13	1294.64	14	1227.62
Rdp104	10	1015.41	10	1000.65	10	**984.81**	10	**984.81**	10	**984.81**	10	1003.58
Rdp105	15	**1375.31**	14	1399.81	14	1377.11	14	1377.11	14	1377.11	14	1383.06
Rdp106	13	1255.48	12	1275.69	12	1261.40	12	**1252.03**	12	**1252.03**	12	1281.52
Rdp107	11	1087.95	11	**1082.95**	10	1144.02	10	1121.86	10	1124.90	11	1092.64
Rdp108	10	967.49	10	**962.48**	9	968.32	9	965.54	9	965.22	10	976.38
Rdp109	12	**1160.00**	12	1181.92	11	1224.86	11	1194.73	11	1203.97	11	1196.88
Rdp110	12	1116.99	11	1106.52	11	**1101.33**	10	1148.20	10	1166.47	11	1127.55
Rdp111	11	1065.27	11	1073.62	10	1117.76	10	1098.84	10	1098.84	11	**1063.06**
Rdp112	10	974.03	10	966.06	10	961.29	9	1010.42	10	**953.63**	10	1025.84
Rdp201	4	1280.44	4	1296.55	4	1254.57	4	1253.23	4	1252.37	5	**1236.13**
Rdp202	4	1100.92	4	1150.31	3	1202.27	3	1191.70	3	1223.69	4	**1098.90**
Rdp203	3	950.79	3	997.84	3	949.42	3	946.28	3	**939.58**	3	946.77
Rdp204	3	775.23	2	848.01	2	837.13	2	833.09	2	835.28	3	**771.89**
Rdp205	3	1064.43	3	1046.06	3	1027.49	3	**994.43**	3	**994.43**	3	1002.98
Rdp206	3	961.32	3	959.94	3	938.63	3	913.68	3	**906.14**	3	927.11
Rdp207	3	835.01	2	899.82	2	912.26	2	890.61	3	**811.51**	3	852.35
Rdp208	3	**718.51**	2	739.06	2	737.26	2	726.82	2	726.83	2	798.14
Rdp209	3	930.26	3	947.80	3	940.29	3	909.16	3	909.16	4	**894.88**
Rdp210	3	983.75	3	1005.11	3	945.97	3	**939.37**	3	939.37	3	978.89
Rdp211	3	839.61	3	812.44	3	805.22	2	904.44	3	**767.82**	3	796.67

Table 5
Experimental results are compared with advanced algorithms for Cdp instance.

Instance	CO-GA		P-SA		VNS-BTST		ALNS-PR		MATE		ACO-DR	
	NV	TD	NV	TD	NV	TD	NV	TD	NV	TD	NV	TD
Cdp101	11	1001.97	11	992.88	11	976.04	11	976.04	11	976.04	11	**970.30**
Cdp102	10	961.38	10	955.31	10	942.45	10	**941.49**	10	**941.49**	10	964.56
Cdp103	10	897.65	10	958.66	10	896.28	10	**892.98**	10	**892.98**	10	913.46
Cdp104	10	878.93	10	944.73	10	872.39	10	**871.40**	10	**871.40**	10	1043.21
Cdp105	11	**983.10**	11	989.86	10	1080.63	10	1053.12	10	1074.51	11	989.59
Cdp106	11	**878.29**	11	**878.29**	10	963.45	10	967.71	10	963.45	11	880.01
Cdp107	11	913.81	11	**911.90**	10	987.64	10	987.64	10	988.60	11	953.14
Cdp108	10	951.24	10	1063.73	10	934.41	10	932.88	10	932.49	10	**930.31**
Cdp109	10	940.49	10	947.90	10	**909.27**	10	910.95	10	**909.27**	10	934.43
Cdp201	3	**591.56**	3	**591.56**	3	**591.56**	3	**591.56**	3	**591.56**	3	**591.56**
Cdp202	3	**591.56**	3	**591.56**	3	**591.56**	3	**591.56**	3	**591.56**	3	**591.56**
Cdp203	3	**591.17**	3	**591.17**	3	**591.17**	3	**591.17**	3	**591.17**	3	**591.17**
Cdp204	3	**590.60**	3	594.07	3	599.33	3	**590.60**	3	**590.60**	3	**590.60**
Cdp205	3	**588.88**	3	**588.88**	3	**588.88**	3	**588.88**	3	**588.88**	3	**588.88**
Cdp206	3	**588.49**	3	**588.49**	3	**588.49**	3	**588.49**	3	**588.49**	3	**588.49**
Cdp207	3	**588.29**	3	**588.29**	3	**588.29**	3	**588.29**	3	**588.29**	3	**588.29**
Cdp208	3	**588.32**	3	**588.32**	3	**588.32**	3	**588.32**	3	**588.32**	3	**588.32**

P-SA algorithm, VNS-BSTS algorithm and ALNS-PR algorithm in solving the TD values, which is the same as MATE algorithm.

Secondly, for the instances where customer location is centralized (Cdp), the results obtained by comparing ACO-DR with advanced algorithms are shown in Table 5.

By analyzing the data in Table 5, it can be concluded that in instances where the customer location is centralized, CO-GA algorithm has 14 NV values and 10 TD values which are the same or better than other algorithms. P-SA algorithm has 14 NV values and 9 TD values which are the same or better than other algorithms. VNS-BSTS algorithm has 17 NV values and 8 TD values which are the same and better than other algorithms. ALNS-PR algorithm has 17 NV values and 11 TD values which are the same and better than other algorithms. MATE algorithm has 17 NV values and 12 TD values which are the same and better than other algorithms. ACO-DR algorithm has 12 NV values and 9 TD values which are the same and better than other algorithms. It can be found through Table 5 that when solving the type 2 instance of Cdp, the six advanced algorithms obtain the same results.

Finally, for the instance where the customer location is a random and centralized combination (RCdp), the results obtained by ACO-DR and the advanced algorithm are shown in Table 6.

Through the analysis of the data in Table 6, it can be seen that for RCdp instance, CO-GA algorithm has 2 NV values and 3 TD values that are superior to other algorithms. P-SA algorithm has 7 NV values and 1 TD values that are superior to other algorithms. VNS-BSTS algorithm has 12 NV values that are superior to other algorithms but no good TD values. ALNS-PR algorithm has 11 NV values and 4 TD values that are superior to other algorithms. MATE algorithm has 12 NV values and 7 TD values that are superior to other algorithms. ACO-DR has 7 NV values and 4 TD values that are superior to other algorithms. Table 6 shows that ACO-DR has no advantage in solving NV values, but it is a relatively good algorithm in solving TD values, because its experimental results are better than CO-GA algorithm, P-SA algorithm and VNS-BSTS, and obtains similar results to ALNS-PR algorithm, but inferior to MATE algorithm.

4.3.3. Comparison with state-of-the-art algorithms on Gehring-Homberge instances

In addition to comparing ACO-DR on Solomon instances, we also test it on Gehring-Homberge instances, which are obtained by extending Solomon's VRPTW benchmark. Then Wang et al.

Table 6
Experimental results are compared with advanced algorithms for RCdp instance.

Instance	CO-GA		P-SA		VNS-BTST		ALNS-PR		MATE		ACO-DR	
	NV	TD	NV	TD	NV	TD	NV	TD	NV	TD	NV	TD
RCdp101	15	**1652.90**	15	1659.59	14	1708.21	14	1776.58	14	1708.21	15	1658.58
RCdp102	14	**1497.05**	13	1522.76	13	1526.36	12	1583.62	12	1570.28	13	1531.25
RCdp103	12	1338.76	11	1344.62	11	1336.05	11	1283.52	11	**1282.53**	12	1312.07
RCdp104	11	1188.49	10	1268.43	10	1177.21	10	1171.65	10	**1171.37**	11	1183.80
RCdp105	14	1571.26	14	1581.54	14	1584.19	14	1584.96	13	1646.36	14	**1569.55**
RCdp106	13	1422.87	13	1418.16	12	1408.19	12	**1392.47**	12	**1392.47**	13	1420.70
RCdp107	12	1282.10	11	1360.17	11	1295.43	11	1255.06	11	**1252.79**	12	1274.64
RCdp108	11	1175.04	11	**1169.57**	10	1207.60	10	1198.36	10	1208.28	11	1171.93
RCdp201	4	1587.92	4	1513.72	4	1437.78	4	1406.94	4	1406.94	4	**1383.79**
RCdp202	4	1211.12	4	1273.26	3	1412.52	3	1414.55	4	**1161.29**	4	1197.36
RCdp203	4	**964.65**	3	1123.58	3	1064.95	3	1050.64	3	1056.96	4	984.59
RCdp204	3	822.02	3	897.14	3	813.74	3	**798.46**	3	**798.46**	3	809.53
RCdp205	4	1410.18	4	1371.08	4	1316.06	4	**1297.65**	4	**1297.65**	4	1370.54
RCdp206	3	1176.85	3	1166.88	3	1154.26	3	1146.32	3	1146.32	3	**1137.28**
RCdp207	4	1036.59	3	1089.85	3	1098.64	3	1061.84	3	1061.14	4	**1032.91**
RCdp208	3	878.57	3	862.89	3	843.30	3	**828.14**	3	828.44	3	862.66

Table 7
Comparison of ACO-DR and advanced algorithms for large instances.

Instance	Size	P-SA		ALNS-PR		VNS-BSTS		ACO-DR	
		NV	TD	NV	TD	NV	TD	NV	TD
C1-2-1	200	21	3169.52	20	2846.20	20	2846.20	20	2704.57
C2-2-1	200	6	1972.87	6	1931.44	6	1931.44	6	1931.44
R1-2-1	200	22	5083.39	20	4849.80	20	4856.09	20	5039.18
R2-2-1	200	5	4372.17	5	4042.67	6	3890.34	5	4354.58
RC1-2-1	200	20	3865.18	19	3652.18	19	3668.98	20	3686.01
RC2-2-1	200	4	2662.75	4	2021.49	6	2165.45	7	2161.58
C1-4-1	400	42	8135.35	40	7533.03	40	7533.14	41	7333.51
C2-4-1	400	14	5085.08	12	4111.84	12	4144.84	15	5149.23
R1-4-1	400	42	12 202.62	40	10 671.70	40	10 591.54	40	10 630.84
R2-4-1	400	9	14 119.64	9	8952.24	11	8745.02	9	9055.01
RC1-4-1	400	40	10 036.82	38	9772.56	39	9020.26	42	9751.30
RC2-4-1	400	13	7229.22	12	6621.94	12	6901.40	15	6928.78
C1-6-1	600	69	19 720.65	63	15 594.21	62	15 916.06	65	16 304.33
C2-6-1	600	20	9509.15	18	7830.16	18	7830.16	23	9348.63
R1-6-1	600	62	25 729.28	59	22 306.17	59	22 811.41	60	22 660.77
R2-6-1	600	13	27 294.11	13	17 459.41	16	18 870.81	15	19 561.77
RC1-6-1	600	60	20 535.26	57	19 679.75	59	18 338.05	62	19 627.08
RC2-6-1	600	20	22 837.36	16	12 693.19	17	13 264.62	20	14 212.54
C1-8-1	800	88	32 801.92	82	27 035.71	82	27 344.05	92	32 035.77
C2-8-1	800	27	14 573.93	24	11 759.05	24	11 957.11	25	17 181.73
R1-8-1	800	93	51 949.49	80	39 348.17	80	39 664.86	83	48 575.97
R2-8-1	800	19	48 611.60	18	27 270.04	22	29 294.91	20	27 177.76
RC1-8-1	800	88	32 801.92	75	38 431.09	78	32 692.26	82	37 565.25
RC2-8-1	800	–	–	60	26 652.10	62	29 640.17	61	29 561.77
C1-10-1	1000	110	52 328.78	102	44 764.64	104	45 303.96	108	53 102.94
C2-10-1	1000	33	23 981.11	30	17 088.50	32	17 869.93	41	27 171.64
R1-10-1	1000	115	77 993.32	100	58 912.62	101	58 208.03	109	64 572.68
R2-10-1	1000	22	67 441.51	22	42 117.48	28	45 533.11	32	47 365.46
C1-10-1	1000	102	66 883.49	93	63 953.66	96	51 345.98	104	60 785.05
RC2-10-1	1000	–	–	75	40 643.56	78	46 073.03	78	46 801.96

(2015) modified this benchmark instance to obtain a large instance set to test VRPSPDTW, which generated 200, 400, 600, 800, and 1000 customer instances. In the current literature on solving VRPSPDTW problem, only three literatures consider the case of large instances and test them with algorithms in the literature, which are P-SA (2015) [5], ALNS-PR (2019) [6] and VNS-BSTS (2020) [7]. The results obtained by ACO-DR and the three algorithms for large instance are shown in Table 7.

By analyzing the data in Table 7, firstly, ACO-PR is compared with P-SA in the 28 benchmark instances, ACO-PR obtained 10 NV values better than P-SA, 7 NV values are the same, and 9 NV values are worse than P-SA's. In terms of TD value, ACO-PR has better TD value than P-SA in 24 out of 28 instances, and only has worse TD value in 4 instances. Since the objective function established by P-SA and ACO-PR is the same when solving VRPSPDTW, it is meaningful to compare the results of these two algorithms. Based on the above analysis of the results, it is clear that ACO-PR generally obtains better results than P-SA, improving the solution of the benchmark example.

Secondly, comparing ACO-DR with ALNS-PR, the two algorithms obtain 6 instances with equal NV values, and the remaining instances where ACO-DR obtains NV values inferior to ALNS-PR. For the TD value, ACO-DR obtained 8 values better than ALNS-PR, 18 values with a gap of less than 20% with ALNS-PR, and 4 values worse than ALNS-PR.

Finally, ACO-DR is compared with VNS-BSTS. Regarding NV value, 5 instances of ACO-DR are better than VNS-BSTS, 4 instances are the same as VNS-BSTS, and the remaining instances are not as good as VNS-BSTS. For the TD value, 6 instances of ACO-DR are better than VNS-BSTS, 20 instances have a gap with VNS-BSTS within 20%, and the remaining 4 instances are worse than VNS-BSTS. Although ACO-DR does not get better results than ALNS-PR and VNS-BSTS, it gets similar results with advanced algorithms. Therefore, it can be considered that the

Table 8
Statistical test results of ACO-DR with advanced algorithms on Rdp instances.

	Algorithm	Signedrank	z-value	p-value	Sig.	Test statistics	
1	ACO-DR with CO-GA	64	−2.205	0.0244	Yes	N	23
2	ACO-DR with P-SA	66	−2.1899	0.0285	Yes	df	5
3	ACO-DR with ALNS-PR	70	−2.0682	0.0386	Yes	p-value	0.0271
4	ACO-DR with VNS-BSTS	74	2.0074	0.1447	Yes	Sig.	Yes
5	ACO-DR with MATE	76	2.0682	0.0386	Yes		

ACO-DR proposed in this paper is effective in solving VRPSPSTW problem.

4.4. Statistical test

The statistical test is a performance measure, similar to other statistics, mean, standard deviation, and success rate (SR), which are used to evaluate whether the proposed algorithm is superior to the comparison algorithm [33]. A key aspect in the design of evolutionary and swarm intelligence algorithms is the study of their performance, and statistical comparisons are also a key part of drawing reliable conclusions. Therefore, statistical test has become an important tool to evaluate the performance of the proposed algorithm, and it is necessary to apply statistical test to the proposed algorithm to verify the experimental results.

In this section, we analyze the performance evaluation of the algorithms from two perspectives: (1) Analysis of variance (ANOVA) test: the purpose is to test whether there is a significant difference between the six algorithms on multiple benchmarks. (2) Pairwise comparison test: the purpose is to test whether there is a significant difference between the proposed ACO-DR algorithm and the advanced algorithm, in other words, it compares the performance of a pair of algorithms on multiple benchmarks. Here, we define the null hypothesis for the ANOVA test and pairwise comparison test respectively: (1) in the ANOVA test, we used the Friedman test that accomplishes the ANOVA test purpose, and the null hypothesis in the ANOVA test is that there is no significant difference in the performance of the six advanced algorithms on multiple solomon benchmarks. (2) In the pairwise comparison test, we used the wilcoxon signed-rank test with significance level $\alpha = 0.05$, and the null hypothesis is that there is no significant difference in the performance of the proposed algorithm with the advanced algorithm. When the null hypothesis is rejected, three symbols are used in this paper to indicate the performance difference between the ACO-DR and the comparison algorithms. "+" means that the performance of ACO-DR is significantly better than the performance of the comparison algorithm; "=" means that the performance of the ACO-DR is not related to the performance of the comparison algorithm; "−" means that the performance of ACO-DR is significantly worse than the performance of the comparison algorithm.

In Section 4.3, the experimental results analyzed are based on the NV and TD perspectives in order to facilitate the comparison with the results obtained by the comparison algorithm. In this section, the analysis of the results of the statistical tests based on the perspective of the total economic efficiency is performed in order to facilitate the comparison of the performance between algorithms. Firstly, we compare the overall performance of all algorithms and the performance of a pair of algorithms on Rdp examples, and the statistical results are shown in Table 8.

In Table 8, the columns of statistical tests represent the results of the ANOVA tests. The rest of the table shows the results of the pairwise comparison test. Where signedrank represents a signed rank statistic.

From the results of the statistical test of variance in Table 8, $p = 0.0271 < 0.05$ and rejecting the null hypothesis, the performance of the six advanced algorithms on Rdp instances is significantly different. Although the ANOVA test can detect a difference between the entire group of algorithms, it does not measure where the difference is. Therefore, it is necessary to use the pairwise comparison test to determine the difference between a pair of algorithms. As can be seen from Table 8, the p-values obtained from the pairwise comparison test are all less than 0.05. So the null hypothesis is rejected, and significant differences exist between the ACO-DR algorithm and the advanced algorithm. The results of the pairwise comparison test for the ACO-DR algorithm with the advanced algorithm on Rdp instances are shown in Table 9.

As can be seen from Table 9, compared with CO-GA and P-SA, the performance of ACO-DR algorithm is significantly improved in 13 instances on the Rdp type instances. Compared with VNS-BSTS, the performance of ACO-DR is significantly improved in 12 instances. Compared with ALNS-PR, the performance of ACO-DR is significantly improved in 8 instances. Compared with MATE, the performance of ACO-DR is significantly improved in 7 instances. In short, when solving Rdp instances, the performance of ACO-DR algorithm is slightly better than CO-GA, P-SA and ANVs-BSTS, but slightly worse than ALNS-PR and MATE.

Secondly, statistical tests are performed between algorithms on the Cdp instance, and the results are shown in Table 10.

From the results of the statistical test of variance in Table 10, $p = 0.9169 > 0.05$ and accepting the null hypothesis, there is no significant difference in the performance of the six advanced algorithms on the Cdp instances. To determine whether there is a significant difference between a pair of algorithms, the pairwise comparison test is used. As shown in Table 10, the p-values obtained from the pairwise comparison test are greater than 0.05, so the null hypothesis is accepted and there is no significant difference between the ACO-DR algorithm and the advanced algorithm. Therefore, the six algorithms have the same performance when solving Cdp instances.

Finally, statistical tests are performed between algorithms on the RCdp instance, and the results are shown in Table 11.

From the results of the statistical test of variance in Table 11, $p = 0.0484 < 0.05$ and rejecting the null hypothesis, the performance of the six advanced algorithms on RCdp instances is significantly different. The pairwise comparison test is used to verify whether there is a significant difference between ACO-DR with those of the state of the art. As shown in Table 11, the p-values obtained from the pairwise comparison test are all less than 0.05. So the null hypothesis is rejected, and significant differences exist between the ACO-DR algorithm and the advanced algorithm. The results of the pairwise comparison test for the ACO-DR algorithm with the advanced algorithm on RCdp instances are shown in Table 12.

As shown in Table 12, compared with CO-GA and P-SA, the performance of ACO-DR algorithm is significantly improved in 11 instances. Compared with VNS-BSTS, the performance of ACO-DR is significantly improved in 9 instances. Compared with ALNS-PR, the performance of ACO-DR is significantly improved in 7 instances. Compared with MATE, the performance of ACO-DR is significantly improved in 7 instances. In short, when solving RCdp instances, the performance of ACO-DR algorithm is slightly better than CO-GA, P-SA and ANVS-BSTS, but slightly worse than ALNS-PR and MATE.

Table 9
Test results of the pairwise comparison for ACO-DR with advanced algorithms on Rdp instances.

Instance	ACO-DR	CO-GA	P-SA	VNS-BSTS	ALNS-PR	MATE
Rdp101	4020.12	3991.06(−)	4005.96(−)	3985.6(−)	3985.6(−)	3985.6(−)
Rdp102	3572.24	3588.08(+)	3595.50(+)	3584.24(+)	3584.24(+)	3584.24(+)
Rdp103	2959.24	2936.32(−)	2957.54(−)	3057.50(+)	3662.02(+)	3057.28(+)
Rdp104	2367.16	2390.82(+)	2361.30(−)	2329.62(−)	2329.62(−)	2329.62(−)
Rdp105	3270.12	3290.62(+)	3303.62(+)	3258.22(−)	3258.22(−)	3258.22(−)
Rdp106	2995.04	2978.96(−)	2983.38(−)	2954.80(−)	2936.06(−)	2936.06(−)
Rdp107	2581.28	2571.90(−)	2561.84(−)	2648.04(+)	2603.72(+)	2609.80(+)
Rdp108	2312.76	2294.98(−)	2284.96(−)	2260.64(−)	2255.08(−)	2254.44(−)
Rdp109	2861.76	2752.00(−)	2795.84(−)	2845.72(−)	2785.46(−)	2803.94(−)
Rdp110	2687.10	2665.98(−)	2609.04(−)	2598.66(−)	2656.40(−)	2692.94(+)
Rdp111	2522.12	2526.54(+)	2543.24(+)	2595.52(+)	2557.68(+)	2557.68(+)
Rdp112	2411.68	2308.06(−)	2292.12(−)	2282.58(−)	2344.84(−)	2267.26(−)
Rdp201	2652.26	2704.88(+)	2717.10(+)	2653.14(+)	2650.46(−)	2648.74(−)
Rdp202	2341.80	2345.84(+)	2444.62(+)	2512.54(+)	2491.40(+)	2555.38(−)
Rdp203	2001.54	2009.58(+)	2103.68(+)	2006.84(+)	2000.56(−)	1987.16(−)
Rdp204	1651.78	1658.46(+)	1658.46(+)	1746.26(+)	1738.18(+)	1742.56(+)
Rdp205	2149.96	2236.86(+)	2200.12(+)	2162.98(+)	2096.86(−)	2096.86(−)
Rdp206	1962.22	2030.64(+)	2027.88(+)	1985.26(+)	1935.36(−)	1920.28(−)
Rdp207	1812.70	1778.02(−)	1871.64(+)	1896.52(+)	1853.22(+)	1731.02(−)
Rdp208	1668.28	1545.02(−)	1550.12(−)	1546.52(−)	1525.64(−)	1525.64(−)
Rdp209	1933.76	1968.52(+)	2003.60(+)	1988.58(+)	1926.32(−)	1926.32(−)
Rdp210	2065.78	2075.50(+)	2118.22(+)	1999.94(−)	1986.74(−)	1986.74(−)
Rdp211	1695.34	1787.22(+)	1732.88(+)	1718.44(+)	1880.88(+)	1643.64(−)
+/ = /−		13/0/10	13/0/10	12/0/11	8/0/15	7/0/16

Table 10
Statistical test results of ACO-DR with advanced algorithms on Cdp instances.

	Algorithm	Signedrank	z-value	p-value	Sig.	Test statistics	
1	ACO-DR with CO-GA	17	–	0.3223	No	N	17
2	ACO-DR with P-SA	13	–	0.1602	No	df	5
3	ACO-DR with ALNS-PR	13	–	0.1603	No	p-value	0.9169
4	ACO-DR with VNS-BSTS	34	–	0.5566	No	Sig.	No
5	ACO-DR with MATE	39	–	0.2754	No		

Table 11
Statistical test results of ACO-DR with advanced algorithms on RCdp instances.

	Algorithm	Signedrank	z-value	p-value	Sig.	Test statistics	
1	ACO-DR with CO-GA	21	−2.4303	0.0151	Yes	N	16
2	ACO-DR with P-SA	22	−2.3786	0.0174	Yes	df	5
3	ACO-DR with ALNS-PR	28	−2.0684	0.0386	Yes	p-value	0.0484
4	ACO-DR with VNS-BSTS	29	1.9649	0.0494	Yes	Sig.	Yes
5	ACO-DR with MATE	30	2.0166	0.0437	Yes		

Table 12
Test results of the pairwise comparison for ACO-DR with advanced algorithms on RCdp instances.

Instance	ACO-DR	CO-GA	P-SA	VNS-BSTS	ALNS-PR	MATE
RCdp101	3857.16	3845.80(−)	3859.18(+)	3920.42(+)	4057.16(+)	3920.42(+)
RCdp102	3530.50	3498.10(−)	3513.52(−)	3520.72(−)	3599.24(+)	3572.56(+)
RCdp103	3056.14	3109.52(+)	3085.24(−)	3068.10(+)	2963.04(−)	2961.06(−)
RCdp104	2763.60	2772.98(+)	2896.86(+)	2714.42(−)	2703.30(−)	2702.74(−)
RCdp105	3703.10	3666.52(−)	3667.08(−)	3600.38(−)	3673.92(−)	3760.72(+)
RCdp106	3309.40	3313.74(−)	3304.32(−)	3248.38(−)	3216.94(−)	3216.94(−)
RCdp107	2981.28	2996.20(+)	3116.34(+)	2986.86(+)	2906.12(−)	2901.58(−)
RCdp108	2739.86	2746.08(+)	2735.14(−)	2775.20(+)	2756.72(+)	2777.16(+)
RCdp201	2947.58	3319.84(+)	3171.44(+)	3018.96(+)	2957.88(+)	2957.88(+)
RCdp202	2538.72	2566.24(+)	2690.52(+)	2933.04(+)	2937.10(+)	2466.58(−)
RCdp203	2113.18	2073.30(−)	2355.16(+)	2237.90(+)	2209.28(+)	2221.92(+)
RCdp204	1727.06	1752.04(+)	1902.28(+)	1735.48(+)	1704.92(−)	1704.92(−)
RCdp205	2885.08	2964.36(+)	2886.16(+)	2776.12(−)	2739.30(−)	2739.30(−)
RCdp206	2418.56	2461.70(+)	2441.76(+)	2416.72(−)	2400.64(−)	2400.64(−)
RCdp207	2209.82	2217.18(+)	2287.70(+)	2305.28(+)	2231.68(+)	2230.28(+)
RCdp208	1833.32	1865.14(+)	1833.78(+)	1794.60(−)	1764.28(−)	1764.88(−)
+/ = /−		11/0/5	11/0/5	9/0/7	7/0/9	7/0/9

4.5. Convergence analysis of the algorithm

In this section, we provide the iterative convergence graph of ACO-DR solving the VRPSPDTW problem. We randomly selected 5 groups of types of instances and compared ACO-DR in different customer sizes of the same type of customers. The obtained convergence curve is shown in Figs. 8–12.

In Figs. 8–11, we randomly selected four types of instances for convergence analysis of the algorithm, and analyzed the influence of the increase of customer size on it.

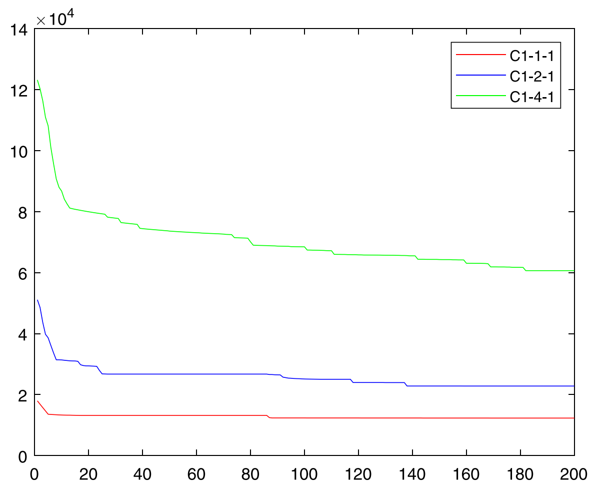

Fig. 8. Convergence curve of C1 type instance.

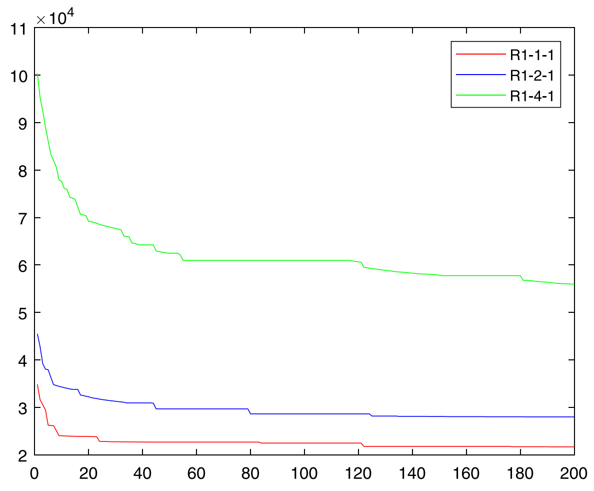

Fig. 9. Convergence curve of R1 type instance.

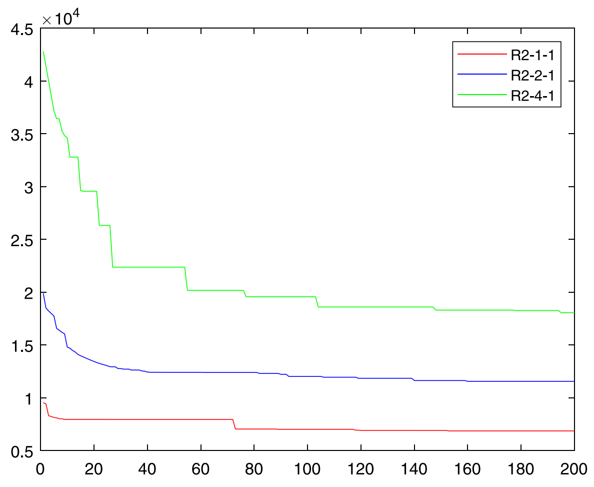

Fig. 10. Convergence curve of R2 type instance.

Fig. 8 shows the convergence curve of C1 instances. It can be seen from Fig. 8 that instances C1-1-1, C1-2-1 and C1-4-1 found the global optimal solution in the middle and later stage of the algorithm.

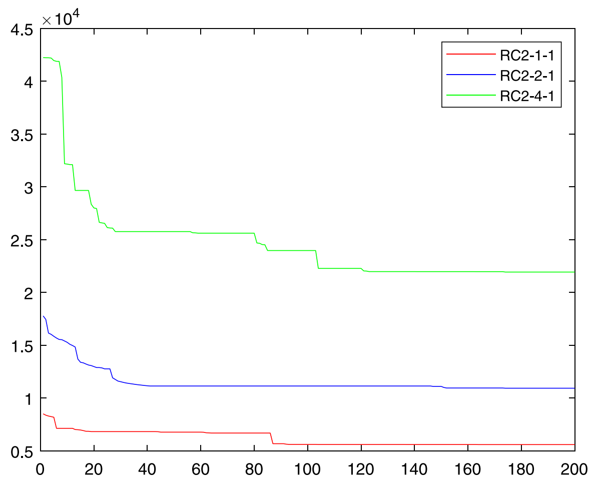

Fig. 11. Convergence curve of RC2 type instance.

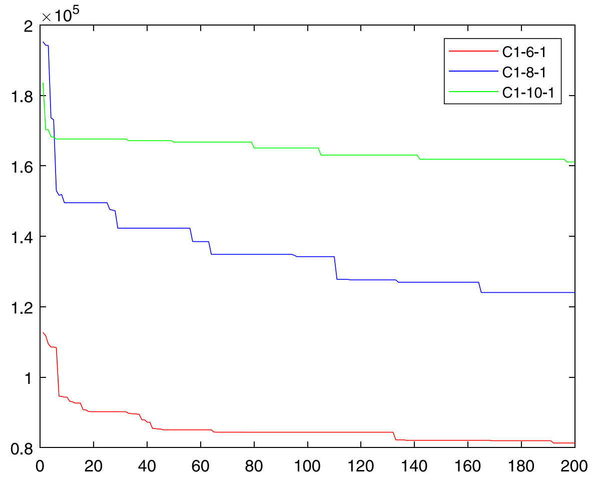

Fig. 12. Convergence curve of C11 type instance.

Fig. 9 shows the convergence curve of R1 instances, and instances R1-1-1, R1-2-1 and R1-4-1 find the global optimal solution when the number of algorithm iterations is 191, 192, 200.

Fig. 10 shows the convergence curve of the R2 type instance. Instances R2-1-1, R2-2-1 and R2-4-1 find the global optimal solution in the middle and late period of algorithm iteration.

Fig. 11 shows the convergence curve of RC2 instances. Instances RC2-1-1, RC2-2-1 and RC2-4-1 find the optimal solution when the number of algorithm iterations is 164, 175, 175.

According to Figs. 8–11, as the number of customers increases, the value of the objective function also increases. The algorithm can constantly find new optimal solutions in the early stage and has good exploration ability. Within a limited number of iterations, the performance of the algorithm also provides the hope of convergence to a better solution in the later stage.

As the size of customers increases, the value of objective function will increase, and the convergence curve of customers with smaller size will not be clearly expressed when compared with instances C1-1-1, C1-2-1 and C1-4-1. Therefore, C1-6-1, C1-8-1 and C1-10-1 are compared separately in Fig. 12.

Fig. 12 shows that the proposed ACO-DR does not stop when solving large-scale problems. As can be seen from the given convergence curve, the convergence speed of the algorithm is fast in the early stage, and then the convergence speed slows down,

Table 13
Distance matrix between distribution center and distribution point.

Node	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	58.01	0														
2	16.28	43.84	0													
3	59.91	106.71	74.97	0												
4	12.21	46.65	11.40	65.19	0											
5	100.69	158.69	115.76	76.22	112.20	0										
6	48.33	92.07	62.36	14.87	51.97	86.31	0									
7	60.46	68.07	50.25	118.60	61.52	134.97	108.41	0								
8	70.18	87.01	63.53	124.26	74.09	129.06	115.88	19.42	0							
9	41.88	99.86	57.25	44.42	53.41	58.82	43.93	85.71	86.38	0						
10	58.67	116.50	74.25	44.55	69.89	42.52	49.50	101.12	99.93	17.09	0					
11	41.04	18.03	28.66	88.96	29.21	141.40	74.43	65.15	82.57	82.62	99.05	0				
12	7.00	54.20	16.495	58.55	7.62	104.81	45.88	64.66	75.6	46.10	62.43	36.62	0			
13	60.21	4.47	45.46	110.32	49.19	160.86	95.75	66.01	85.15	102.08	118.85	21.38	56.80	0		
14	33.30	64.09	28.64	89.44	38.08	109.77	79.88	29.41	36.88	57.04	73.11	53.60	38.83	63.95	0	
15	30.36	46.62	17.69	90.25	29.07	121.85	78.55	32.65	47.17	65.51	82.46	37.01	33.12	46.53	17.46	0

but the exploration ability is not weakened. Therefore, ACO-DR is effective in solving large-scale problems.

4.6. Computational complexity analysis of algorithm

The computational complexity is mainly to the analyze the efficiency of the algorithm. In general, we evaluate the complexity of the algorithm mainly includes two aspects: time complexity and space complexity.

In most of the literature, the time complexity is used to measure the running time of the algorithm. However, this result is highly susceptible to the environment in which the same algorithm is executed at different times on different computers. When comparing the time complexity between different algorithms, it is important to ensure that all algorithms run in the same environment. Because we obtained the best test results of these algorithms directly from the original publication for comparison, it is not rigorous to analyze the algorithms in terms of time complexity. Therefore, the computational complexity of the algorithm is expressed by space complexity in this paper.

Space complexity is used to measure the size of the storage space required by the algorithm during its operation. In the VRPSPDTW problem studied in this paper, there are N+1 nodes (N customer nodes and one depot node). M ants are used in the ACO-DR algorithm, then the storage space required by the algorithm is mainly as follows: (1) the pheromone matrix: Tau[N+1][N+1] is used to store pheromone between N+1 nodes, and its complexity is $O([N+1]^2)$; (2) tabu table: Tabu[M][N+1] is used to store and record the generation of paths, its complexity is $O(M*[N+1])$; (3) distance matrix: D[N+1][N+1] is used to store the distance between N+1 nodes, its complexity is $O([N+1]^2)$; (4) path matrix: L[N+1][1] is used to store the path length searched by each ant in this iteration, and its complexity is $O(N+1)$; (5) the complexity of the optimal solution is $O(M)$; (6) local search: the number of local searches is L_m, and the complexity of the local search is $O(L_m * N)$. Therefore, through comprehensive analysis of ACO-DR algorithm, the space complexity S(N+1) in the whole calculation process is: $S(N+1) = O([N+1]^2) + O(M*[N+1]) + 2O(L_m * N)$.

5. Practical application

In order to verify the feasibility of the solution proposed in this paper for the VRPSPDTW problem in a practical problem, we apply the above mentioned model and the designed algorithm to the distribution problem of logistics company. In this paper, we take a regional distribution of a logistics company in Ningxia as an example, which has the following characteristics in distribution operation: (1) vehicles depart from the logistics center to the corresponding locations within the city for the distribution of goods; (2) the distribution mode is not a simple pickup or delivery, but a pickup and delivery integrated mode, that is, delivery and pickup at the same time; (3) the distribution orders are static, there is no dynamic situation of temporary insertion of emergency orders in the process of vehicle distribution; (4) The logistics service provided by the company is strictly controlled within the time window requested by the customer to prevent the distribution plan from being disrupted and the company's reputation from being damaged. Due to the current order quantity of the company is not many, so the main way is to use manual wiring for vehicle scheduling and goods distribution, but if the number of orders increases in the future, then the solution of using manual wiring may no longer be applicable. Therefore, the company needs to introduce intelligent means in logistics distribution, process orders in time and give the corresponding distribution adjustment scheme, and reduce the distribution cost loss caused by the deviation of manual judgment.

The data selected in this paper are the actual logistics orders of a certain area of the logistics company. There are 15 orders in the distribution center on the day, and the exact same type of vehicles are selected for cargo transportation. The maximum load of the vehicles is 300 kg, the fixed fee for each vehicle is 60 yuan, and the cost of the vehicles per kilometer is 5 yuan. Now we need to provide a reasonable operation plan for the distribution center's vehicles to reduce the company's costs. The distance matrix of distribution center (0) and 15 distribution locations is shown in Table 13, and the specific information of each distribution location (including demand, recovery, time window, etc.) is shown in Table 14.

According to the model established in the second part and the algorithm designed in the third part of this paper, we apply ACO and the improved algorithm ACO-DR to solve this real-world problem respectively, and the results obtained are shown in Tables 15 and 16.

It can be seen from Table 15 that four vehicles are needed to solve the VRPSPDTW instance using ACO. According to the last line, the total distance is 971.80 km and the fleet needs to spend 5098.98 yuan to complete all the distributions. However, three vehicles are needed when ACO-DR is used to solve the VRPSPDTW example in Table 16, the total distance are 775.67 km, which is 196.13 km less than the total distance in Table 15. The cost of the fleet to complete all distributions is 4055.63 yuan and decreases from 1043.35 yuan. Therefore, the application of the example and the analysis of the results show that the algorithm proposed in this paper has good applicability to practical problems.

6. Conclusion

In this paper, an improved ant colony algorithm ACO-DR is proposed for VRPSPDTW. Based on the basic ant colony algorithm,

Table 14
Details of each distribution point.

Point	Pickup (kg)	Delivery (kg)	Left time window	Right time window	Service time (min)
0	0	0	8:00	17:00	0
1	85	70	8:00	10:00	10
2	25	12	12:50	13:40	3
3	36	42	14:30	15:30	5
4	65	55	15:00	17:00	5
5	88	72	10:30	14:00	10
6	76	41	12:40	14:40	5
7	25	45	9:00	10:45	4
8	92	84	10:50	12:50	10
9	78	70	12:40	13:50	9
10	66	60	9:00	11:30	8
11	48	54	10:00	11:00	6
12	74	68	9:30	12:30	5
13	21	13	13:40	14:50	3
14	36	28	14:30	16:20	3
15	53	60	14:15	16:00	4

Table 15
Results of ACO solving VRPSPDTW instances.

Vehicle	Distance (km)	Cost (yuan)	Route
1	367.87	1899.35	0-1-12-11-5-0
2	176.02	940.08	0-2-13-15-14-4-0
3	304.80	1854.00	0-7-10-5-9-0
4	123.11	675.55	0-6-3-0
Total	971.80	5098.98	

Table 16
Results of ACO-DR solving VRPSPDTW instances.

Vehicle	Distance (km)	Cost (yuan)	Route
1	266.64	1390.20	0-5-9-6-3-0
2	276.39	1441.95	0-10-12-11-13-15-14-0
3	232.64	1223.20	0-1-7-8-2-4-0
Total	775.67	4055.35	

the time urgency and service time of customers are considered as the factors affecting the state transition, and a random transition rule with direction is constructed to improve the probability of the algorithm to search the target. In the process of finding the optimal path, the destory operator and repair operator are added to avoid the algorithm falling into the local optimum. Through a large number of numerical experiments on the proposed algorithm and comparing it with sate-of-the-art algorithms, the following conclusions can be drawn: (1) considering the particularity of the problem (considering practical factors), the ACO-DR algorithm proposed in this paper can obtain a good value when solving TD, while the result is not satisfactory when solving NV. (2) According to the convergence curve, the strategy proposed in this paper is effective and improves the exploration ability of the algorithm, without the stagnation of the algorithm. (3) Numerical experiments prove that the ACO-DR is an effective method for solving VRPSPDTW, which provides a new idea and solution for solving the VRPSPDTW problem. Finally, the mathematical model and the proposed algorithm are applied to the real logistics distribution, which proves that the model and algorithm are feasible and effective in practical problems.

In the future research, we will change the target from single-depot to multi-depot. To provide effective solutions for different types of vehicle routing problems, specific types of VRP people can solve it with the help of automated methods to get their own satisfactory results.

Funding

This work was supported by the Key Project of Ningxia Natural Science Foundation (2022AAC02043), the Construction Project of First-class Subjects in Ningxia Higher Education (NXYLXK2017-B09) and the Major Proprietary Funded Project of North Minzu University (ZDZX201901), the Basic Discipline Research Projects supported by Nanjing Securities (NJZQJCXK202201).

CRediT authorship contribution statement

Hongguang Wu: Conceptualization, Methodology, Software, Writing – review & editing. **Yuelin Gao:** Supervision, Writing – original draft, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

https://neo.lcc.uma.es/vrp/vrp-instances/ or https://www.bernabe.dorronsoro.es/vrp.

References

[1] H. Min, The multiple vehicle routing problem with simultaneous delivery and pick-up points, Transp. Res. A 1 (1989) 377–386.
[2] M.Y. Lai, E.B. Cao, An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time window, Eng. Appl. Artif. Intell. 23 (2010) 188–195, http://dx.doi.org/10.1016/j.engappai.2009.09.001.
[3] H.F. Wang, Y.Y. Chen, A genetic algorithm for the simultaneous delivery and pickup problems with time window, Comput. Ind. Eng. 62 (2012) 84–95, http://dx.doi.org/10.1016/j.cie.2011.08.018.
[4] C. Wang, F. Zhao, D. Mu, J.W. Sutherland, Simulated annealing for a vehicle routing problem with simultaneous pickup-delivery and time windows, in: IFIP International Conference on Advances in Production Management Systems.
[5] C. Wang, F. Zhao, D. Mu, J.W. Sutherland, A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup-delivery and time windows, Comput. Ind. Eng. 83 (2015) 111–122, http://dx.doi.org/10.1016/j.cie.2015.02.005.
[6] J. Hof, M. Schneider, An adaptive large neighborhood search with path relinking for a class of vehicle-routing problems with simultaneous pickup and delivery, Networks 74 (2019) 207–250, http://dx.doi.org/10.1002/net.21879.
[7] Y. Shi, Y.J. Zhou, T.F. Boudouh, O. Grunder, A lexicographic-based two-stage algorithm for vehicle routing problem with simultaneous pickup-delivery and time window, Eng. Appl. Artif. Intell. 95 (2020) 103901, http://dx.doi.org/10.1016/j.engappai.2020.103901.
[8] S.C. Liu, K. Tang, X. Yao, Memetic search for vehicle routing with simultaneous pickup-delivery and time windows, Swarm Evol. Comput. 66 (2021) 100927, http://dx.doi.org/10.1016/j.swevo.2021.100927.
[9] J.H. Wang, Y. Zhou, Y. Wang, J. Zhang, C.L.P. Chen, Z.B. Zheng, Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: formulation, instances, and algorithms, IEEE Trans. Cybern. 46 (2015) 582–594, http://dx.doi.org/10.1109/TCYB.2015.2409837.
[10] H.Y. Li, L. Wang, X.H. Hei, W. Lia, Q.Y. Jiang, A decomposition-based chemical reaction optimization for multi-objective vehicle routing problem for simultaneous delivery and pickup with time windows, Memet. Comput. 10 (2018) 103–120, http://dx.doi.org/10.1007/s12293-016-0222-1.
[11] Y. Wang, L.Y. Ran, X.Y. Guan, J.X. Fan, Y.Y. Sun, H.Z. Wang, Collaborative multicenter vehicle routing problem with time windows and mixed deliveries and pickups, Expert Syst. Appl. 197 (2022) 116690, http://dx.doi.org/10.1016/j.eswa.2022.116690.
[12] R. Liu, X.L. Xie, V. Augusto, C. Rodriguez, Heuristic approaches for a special simultaneous pickup and delivery problem with time windows in home health care industry, IFAC Proc. Vol. 45 (2012) 345–350, http://dx.doi.org/10.3182/20120523-3-RO-2023.00204.
[13] R. Liu, X.L. Xie, V. Augusto, C. Rodriguez, Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care, European J. Oper. Res. 230 (2013) 475–486, http://dx.doi.org/10.1016/j.ejor.2013.04.044.
[14] M. Chaieb, D.B. Sassi, Measuring and evaluating the home health care scheduling problem with simultaneous pick-up and delivery with time window using a tabu search metaheuristic solution, Appl. Soft Comput. 113 (2021) 107957, http://dx.doi.org/10.1016/j.asoc.2021.107957.

[15] A. Colorni, M. Dorigo, V. Maniezzo, Distributed optimization by ant colonies, in: Proceedings of ECAL91 - European Conference on Artificial Life, 1991, pp. 134–142.

[16] L. Xua, K. Huang, J.P. Liu, D.S. Li, Y.F. Chen, Intelligent planning of fire evacuation routes using an improved ant colony optimization algorithm, J. Build. Eng. (2022) 105208, http://dx.doi.org/10.1016/j.jobe.2022.105208.

[17] Z.R. Dong, X.Y. Bian, S. Zhao, Ship pipe route design using improved multi-objective ant colony optimization, Ocean Eng. 258 (2022) 111789, http://dx.doi.org/10.1016/j.oceaneng.2022.111789.

[18] Y. Chen, M.Q. Chen, Z.H. Chen, Cheng, Y.H. Yang, H. Li, Delivery path planning of heterogeneous robot system under road network constraints, Comput. Electr. Eng. 92 (2022) 107197, http://dx.doi.org/10.1016/j.compeleceng.2021.107197.

[19] A. Kumar, M. Thakur, G. Mittal, Planning optimal power dispatch schedule using constrained ant colony optimization, Appl. Soft Comput. 115 (2022) 108132, http://dx.doi.org/10.1016/j.asoc.2021.108132.

[20] R. Skinderowicz, Improving ant colony optimization efficiency for solving large TSP instances, Appl. Soft Comput. 120 (2022) 108653, http://dx.doi.org/10.1016/j.asoc.2022.108653.

[21] Y. Wang, Z. Han, Ant colony optimization for traveling salesman problem based on parameters optimization, Appl. Soft Comput. 107 (2021) 107439, http://dx.doi.org/10.1016/j.asoc.2021.107439.

[22] N.A. Kyriakakis, M. Marinaki, Y. Marinakis, A hybrid ant colony optimization-variable neighborhood descent approach for the cumulative capacitated vehicle routing problem, Comput. Oper. Res. 134 (2021) 105397, http://dx.doi.org/10.1016/j.cor.2021.105397.

[23] X.S. Xiang, J.F. Qiu, J.H. Xiao, X.Y. Zhang, Demand coverage diversity based ant colony optimization for dynamic vehicle routing problems, Eng. Appl. Artif. Intell. 91 (2020) 103582, http://dx.doi.org/10.1016/j.engappai.2020.103582.

[24] Y.B. Li, H. Soleimani, M. Zohal, An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives, J. Clean. Prod. 227 (2019) 1161–1172, http://dx.doi.org/10.1016/j.jclepro.2019.03.185.

[25] N. Guo, B. Qian, J. Na, R. Hu, J. Mao, A three-dimensional ant colony optimization algorithm for multi-compartment vehicle routing problem considering carbon emissions, Appl. Soft Comput. 127 (2022) 109326, http://dx.doi.org/10.1016/j.asoc.2022.109326.

[26] J.C. Molina, J.L. Salmeron, I. Eguia, An ACS-based memetic algorithm for the heterogeneous vehicle routing problem with time window, Expert Syst. Appl. 157 (2020) 113379, http://dx.doi.org/10.1016/j.eswa.2020.113379.

[27] H.Z. Zhang, Q.W. Zhang, L. Ma, Z.Y. Zhang, Y. Liu, A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows, Inform. Sci. 490 (2019) 166–190, http://dx.doi.org/10.1016/j.ins.2019.03.070.

[28] Q.L. Ding, X.P. Hu, L.J. Sun, Y.Z. Wang, An improved ant colony optimization and its application to vehicle routing problem with time windows, Neurocomputing 98 (2012) 101–107, http://dx.doi.org/10.1016/j.neucom.2011.09.040.

[29] Y. Gajpal, P. Abad, An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup, Comput. Oper. Res. 36 (2009) 3215–3223, http://dx.doi.org/10.1016/j.cor.2009.02.017.

[30] C.B. Kalayci, C. Kaya, An ant colony system empowered variable neighborhood search algorithm for the vehicle routing problem with simultaneous pickup and delivery, Expert Syst. Appl. 66 (2016) 163–175, http://dx.doi.org/10.1016/j.eswa.2016.09.017.

[31] W.J. Liu, Y.T. Zhou, W. Liu, J. Qiu, N.M. Xie, X.Y. Chang, J. Chen, A hybrid ACS-VTM algorithm for the vehicle routing problem with simultaneous delivery & pickup and real-time traffic condition, Comput. Ind. Eng. 162 (2021) 107747, http://dx.doi.org/10.1016/j.cie.2021.107747.

[32] S.F. Ghannadpour, S. Noori, R. Tavakkoli-Moghaddam, K. Ghoseiri, A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application, Appl. Soft Comput. 14 (2014) 504–527, http://dx.doi.org/10.1016/j.asoc.2013.08.015.

[33] J. Carrasco, S. Garci, M.M. Rueda, S. Das, F. Herrera, Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review, Swarm Evol. Comput. 54 (2020) 100665, http://dx.doi.org/10.1016/j.swevo.2020.100665.