Complex & Intelligent Systems
https://doi.org/10.1007/s40747-021-00401-1

ORIGINAL ARTICLE

A hybrid ant colony algorithm based on multiple strategies for the vehicle routing problem with time windows

Hongguang Wu[1,2] · Yuelin Gao[1,3] · Wanting Wang[2,3] · Ziyu Zhang[2,3]

Received: 9 January 2021 / Accepted: 13 May 2021

Abstract

In this paper, we propose a vehicle routing problem with time windows (TWVRP). In this problem, we consider a hard time constraint that the fleet can only serve customers within a specific time window. To solve this problem, a hybrid ant colony (HACO) algorithm is proposed based on ant colony algorithm and mutation operation. The HACO algorithm proposed has three innovations: the first is to update pheromones with a new method; the second is the introduction of adaptive parameters; and the third is to add the mutation operation. A famous Solomon instance is used to evaluate the performance of the proposed algorithm. Experimental results show that HACO algorithm is effective against solving the problem of vehicle routing with time windows. 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.

Keywords Vehicle routing problem · Ant colony algorithm · Mutation operation · Adaptive parameter · Practical application

Introduction

The twenty-first century is an information age, information technology and scientific research promote the development of the world, the process of globalization continues to deepen, the rapid development of the economy cannot be separated from the public consumption, which makes the development of the service industry is faster and faster, and the problem of vehicle routing is also proposed. Vehicle routing problem (VRP) refers to the need for a group of vehicles to find an economical and effective route to meet the needs of different customers. It is a famous NP-hard combination problem in logistics management, vehicle scheduling, and traffic transportation [1]. The problem of vehicle routing was first mentioned in 1959 by Dantzig and Ramser. With the question raised, many variations on VRP have emerged. Such as VRP with capacity (CVRP) [2], VRP with time windows (TWVRP) [3,4], VRP with fleet size (FSVRP) [5], and VRP with uncertain requirements [6], these extended VRP also belongs to NP-hard combination problems.

The simplest routing problem is the traveling salesman problem (TSP); if we put limits to the carrying capacity, then this problem turns into the capacitated vehicle routing problem (CVRP) [2]. If we limit the distance between vehicles for CVRP, then it becomes VRP with distance constraint, namely DVRP [2]. In DVRP, if we consider the time of serving the demand of customer, the problem becomes TWVRP that we want to discuss [3].

Ombuki et al. [7] based on the Pareo sorting technique for the genetic algorithm to solve the TWVRP problem, they presented the TWVRP as a multi-objective problem.

This research is supported by the National Natural Science Foundation of China under Grant (11961001, 61561001), the Construction Project of First-class Subjects in Ningxia Higher Education (NXYLXK2017B09), and the Major Proprietary Funded Project of North Minzu University (ZDZX201901).

✉ Yuelin Gao
gaoyuelin@263.net

Hongguang Wu
2440576496@qq.com

Wanting Wang
874812086@qq.com

Ziyu Zhang
2541604895@qq.com

[1] School of Mathematics and Information Science, North Minzu University, Yinchuan 750021, China

[2] Ningxia Province Key Laboratory of Intelligent Information and Data Processing, North Minzu University, Yinchuan 750021, China

[3] Ningxia Province Cooperative Innovation Center of Scientific Computing and Intelligent in Formation Processing, North Minzu University, Yinchuan 750021, China

Published online: 01 June 2021

The effectiveness of the proposed algorithm is verified by comparing with baseline data. Ghoseiri et al. [8] combined genetic algorithm with evolutionary search and Pareo technology to get an adaptive genetic algorithm for solving the TWVRP. They converted the TWVRP into a goal programming and got good results. Similar algorithms can also be used to solve such problems, such as the differential evolutionary integration algorithm based on mixed penalty function [9] and the differential evolutionary algorithm based on topology [10] proposed by Wang.

Ho et al. [11] used tabu search algorithm when solving the TWVRP, and the experimental results showed that this problem could be solved by tabu search algorithm and satisfactory results were obtained. Belhaiza et al. [12] proposed a tabu search algorithm combined with variable neighborhood method to solve the TWVRP. The advantage of this algorithm is that the minimum waiting time and the minimum delay of customers can be recorded in the process of path generation, and then adjust the arrival times and departure times accordingly. Wang et al. [13] proposed a threshold tabu search algorithm to solve the TWVRP, which can make vehicle dispatchers respond to practical problems effectively and quickly.

Chiang et al. [14] proposed a hybrid simulated annealing algorithm to solve the TWVRP, and combined with tabu table to increase the annealing process of short-term memory function. Numerical experiments showed that this method is better than the previous research results. Deng et al. [15] improved the simulated annealing algorithm by adding memory function, and adopted the double termination principle to solve the TWVRP. The experimental results also confirmed that the new algorithm could obtain better solutions. Wang et al. [16] used the improved parallel simulated annealing algorithm based on residual capacity and radial additional insertion to solve the pick up and deliver at the same time within a specific time window (VRPSPDTW), and minimized the target with a mixed integer programming model.

Yu et al. [17] accumulated heuristic information using pheromone matrix method to obtain the IACO when solving the period vehicle routing problem with time windows (PTWVRP), and introduced double crossover operation in the algorithm to improve the performance of the algorithm. Ding et al. [18] proposed a hybrid ant colony optimization algorithm by introducing disaster operator and adjusting pheromone. Wang et al. [19] treated the TWVRP as a multi-objective problem and proposed a hybrid ant colony optimization algorithm, which is obtained by combining ant colony algorithm and simulated annealing. The performance of the improved algorithm was significantly improved. In addition, these algorithms can also be used to solve vehicle routing problems, such as particle swarm optimization algorithm [20–23], bee colony algorithm [24–26], whale optimization algorithm [27], Dijkstra algorithm [28], and so on.

Ant colony optimization (ACO) is an effective heuristic algorithm to solve the combination optimization problems, which provides a new idea about solving the VRP. Clolrni et al. [29] proposed ant colony algorithm, which was proposed on the basis of real ants and successfully applied to traveling salesman problem. Bulleneimer et al. [30] first proposed the ant system, they improved the algorithm and applied it to the VRP. Due to the complexity and extensiveness of reality, the basic ant colony algorithm has limitations on solving practical problems. Therefore, many researchers proposed new methods and new ideas to improve the ant colony algorithm and expand the application scope of the ant colony algorithm. Yu et al. [31] changed the pheromone update mode, introduced the mutation operation to obtain the IACO. Although the improved algorithm avoids the local optimization, it greatly increases the complexity of the algorithm. Mavrovouniotis et al. [32] combined ant colony algorithm with local search algorithm to solve the dynamic traveling salesman problem. This algorithm improves the quality of the solution, but its application effect on dynamic instances is poor. Ding et al. [18] proposed a hybrid ant colony algorithm and introduced the disaster operator. They studied the convergence speed of the proposed algorithm, but does not consider the complexity of the algorithm. Wang et al. [33] applied the hybrid ant colony algorithm to the model of emergency transportation after a disaster. Jabir et al. [34] combined ant colony algorithm with variable neighborhood search algorithm and applied it to multi-stage green vehicle routing problem (MDGVRP). Li et al. [35] applied the improved ant colony algorithm to multi-target vehicle routing problem (MVRP). He proposes a relatively simple model and only compares the proposed algorithm with the basic algorithm when evaluating its performance.

Therefore, this paper proposes a HACO algorithm (HACO) based on multiple strategies to solve the TWVRP. The main work are to improve the updating method of pheromone , adjust the pheromone volatile factor adaptively, and introduce into swap operator and insert operator to avoid the algorithm falling into local optimal and improve the convergence rate of the algorithm. The framework of this paper is divided into six parts: the first part is the introduction; the second part is the problem modeling; the third part is the solution of the problem; the fourth part is the numerical experiment and the result analysis; the fifth part is an example application; the sixth part is the conclusion.

Problem description and formulation

The TWVRP requires a group of vehicles to start from the depot and complete customer service requirements within the time interval. And on this basis to find a shortest distance and effective route. The TWVRP is widely used in logistics management and the most common problem with transportation system.

We can use the graph theory to model TWVRP as a directed complete graph $G = (V, E)$ [37], with a vertex set $V = \{0, 1, 2, ..., n\}$ and an edge set $E = \{(i, j)|i, j \in V\}$. In TWVRP, each vertex represents the node (or customer), and each edge represents the distance between two nodes. In this problem, $M = \{1, 2, ..., m\}$ represents a group of vehicles. We assume that the depot is vertex 0 and customers are introduces as nodes. The vehicle k $\{k \in M\}$ needs to start from the depot and return to the depot after serving a certain number of node. For nodes that is requirements, only one vehicle can serve them and cannot exceed the maximum capacity of the vehicle. In addition, the vehicle needs to service within the time interval$[a, b]$; it is not earlier than the left time window a or later than the right time window b.

In this paper, we consider the constraints on vehicle capacity and time window to reduce vehicle use and travel distance and minimize the costs associated with it.

For the convenience of the following description, we define it as follows:

a_i : The left time window of the node i
b_i : The right time window of the node i
c_{ij} : Distance between node i and node j
Cap_k : The capacity of vehicle k
d_i : The demand of node i
g_k : The fixed cost incurred for using vehicle k
h_k : The transportation cost per unit distance of vehicle k
t_{ij} : The time required for traveling from node i to node j
s_i : The service time of node i
r_i : The time when vehicle arrives at node i
Decision variables

$$x_{ijk} = \begin{cases} 1, & if\ vehicle\ k\ travels\ from\ node\ i\ to\ node\ j \\ 0, & otherwise \end{cases} \tag{1}$$

$$y_{ik} = \begin{cases} 1, & if\ node\ i\ is\ served\ by\ vehicle\ k \\ 0, & otherwise. \end{cases} \tag{2}$$

Objectives: minimum cost

Constraints

The total demand cannot exceed the capacity of the vehicle.

Each node has a vehicle to serve and can only be served by one vehicle.

The vehicle must serve the node within the time interval.

The vehicle leaves the depot and finally returns to the depot.

According to the above-defined objectives, parameters, and decision variables, the problem can be represented as

$$\min \quad \sum_{k=1}^{m} h_k \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} x_{ijk} + \sum_{k=l}^{m} g_k \sum_{j=1}^{n} x_{0jk} \tag{3}$$

$$s.t. \quad \sum_{k=1}^{m} d_i y_{ik} \le Cap_k, \quad \forall k \in \{1, 2, \ldots, m\} \tag{4}$$

$$\sum_{k=1}^{m} y_{ik} \le 1, \quad \forall i \in \{1, 2, \ldots, n\} \tag{5}$$

$$\sum_{j=1}^{n} x_{0jk} - \sum_{i=1}^{n} x_{i0k} = 0, \quad \forall k \in \{1, 2, \ldots, m\} \tag{6}$$

$$\sum_{i=0}^{n} x_{ijk} = y_{ik}, \forall j \in \{1, 2, \ldots, n\} \quad \forall k \in \{1, 2, \ldots, m\} \tag{7}$$

$$\sum_{j=0}^{n} x_{ijk} = y_{ik}, \forall i \in \{1, 2, \ldots, n\} \quad \forall k \in \{1, 2, \ldots, m\} \tag{8}$$

$$a_i \le r_i \le b_i, \quad \forall i \in \{1, 2, \ldots, n\} \tag{9}$$

$$\sum_{i=0}^{n} \sum_{i=0}^{n} x_{ijk}(t_{ij} + s_i + r_i) = r_j, \quad \forall k \in \{1, 2, \cdots, m\}. \tag{10}$$

In the above model, objective (3) is to minimize the cost of the vehicle. Constraint (4) ensures that the total demand of nodes cannot exceed the capacity of the vehicle. Constraint (5) ensures that each node can only use one vehicle. Constraint (6) ensures that the vehicle starts and ends in the depot when serving the node. Constraints (7) and (8) ensure that each node always has one vehicle to serve them. Constraint (9) ensures that the service vehicle should arrive within the time period. Constraint (10) ensures that the vehicle finally arrives at the node j from the node i by passing service time and travel time.

Solution approach

The ant colony algorithm is evolved based on the foraging behavior of ants in nature. During the process of searching for food, ants leave secretions along the path they passed. An ant colony can always find a shortest path from its nest to food source through these secretions. Dorigo [38–40] proposed the concept of ACO after being inspired by the foraging behavior of ant colonies. The idea of ACO is to use the secretions left by the ants to exchange information, and then find the optimal path required by the problem. The secretions are called pheromones. In the foraging process, the ant determines the next step based on the distribution of pheromones. If more pheromones are found in a certain path, the probability of the ant choosing this path will be higher. If the ant leaves fewer pheromones, the probability of the ant choosing this path will be lower. So as more and more pheromones are accumulated in the path, it means that more and more ants will choose, and the ant can find the shortest way to find food. As a heuristic algorithm, the ACO is often used to solve various optimization problems, and has been successfully applied to practical cases, especially in VRP, TSP, and related extension problems.

The core of the ACO is to determine the node that the ant chooses to go next and design the pheromone update formula for how to guide the ant to find the shortest path more quickly. The improvement on ant colony algorithm in this paper is based on Dorigo (1992) [36] and Ding et al. (2012) [18], and mainly improves the updating formula of pheromone and adds adaptive parameters. The parameter ρ in pheromone updating formula is changed from constant to dynamic number to reduce the value of the bad path during the cycle, and increase the role of the good path in the optimization process.

The improvement of ACO

When using HACO to solve the TWVRP, each ant needs to build a path from the depot that satisfies the constraints. If it is found that the next node selected in the current path does not satisfy the condition, then the ant needs to return the depot and rebuild the new path with the remaining nodes, and this process stops until all nodes are selected. Each ant through a certain probability to select the next node. Therefore, the formula for the node i to select the next node j is

$$j = \begin{cases} \max_{j \in N_i^k} \left\{ \tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta} \cdot \left(\frac{1}{width_i} \right)^{\gamma} \right\}, & q \leq q_0 \\ P_{ij}^k = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta} \cdot \left(\frac{1}{width_i} \right)^{\gamma}}{\max_{j \in N_i^k} \left\{ \tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta} \cdot \left(\frac{1}{width_i} \right)^{\gamma} \right\}}, & q > q_0 \end{cases} \tag{11}$$

In the formula (11), N_i^k represents the set of node that the vehicle can access after it leaves the node i; τ_{ij} represents the number of pheromone on the path between the current node i and possible node j; η_{ij} represents the intensity, its expression is $\eta_{ij} = \frac{1}{c_{ij}}$, and it is the reciprocal of the distance between the node i and possible node j; $width_i$ represents the width of the time window or the urgency of the node's requirements, and is expressed as follows: $width_i = b_i - a_i$; α is a pheromone factor that determines the ant's trajectory; β is the expectation factor, which reflects the importance of external information to the trajectory; γ represents the importance of the urgency of the service node to trajectory selection. The node j selected is the more urgent, the greater the pheromone concentration and the smaller the distance when $q \leq q_0$; and it is according to the probability formula P_{ij}^k and roulette method to choose node j when $q > q_0$; $q \in rand(0, 1)$ is the random number; q_0 is a parameter that controls the exploitation against the exploration of ant during the search progress.

When determining which node the ant wants to go next, and the equivalent of determining the direction. They leave pheromone on the path to communicate information about their companions. The pheromone left by the former will influence the path chosen by the ants later, and in other words, the pheromone concentration decides the choice of the path. According to the principle of ACO, the shorter the path, the higher the pheromone concentration; the longer the path, the smaller the pheromone concentration. To speed up the search process of ants, the updating formula for pheromone is improved, which is as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho \bigtriangleup \tau_{ij}(t) \tag{12}$$

$$\bigtriangleup \tau_{ij}(t) = \sum_{u=1}^{v} \bigtriangleup \tau_{ij}^{u}(t) \tag{13}$$

$$\bigtriangleup \tau_{ij}^{u} = \begin{cases} \frac{Q}{L_u}, & if\ the\ ant\ u\ goes\ through\ the\ path\ (i, j) \\ 0, & otherwise \end{cases} \tag{14}$$

In the above formula, ρ is the pheromone volatility coefficient; $\bigtriangleup \tau_{ij}^{u}$ represents the number of pheromones on the path of the ant u in the current cycle; Q is a constant related to the amount of pheromone deposited by the ants; L_u represents the length of the ant u path in the current cycle.

The pheromone volatility coefficient ρ is a dynamic value. If the value of ρ is small, the pheromones in the path evaporate more slowly, and the pheromone concentration remaining on the path will be large, which will have a great influence on the subsequent ant's path selection, thus reducing the ant's random search ability; when the value of ρ is large, the pheromone on the path will evaporate quickly, while the concentration of pheromone remaining on the path will be small, which increases the ant's global searching ability in the search process and makes it easier to explore other paths. In the ant search process, to reduce the interference with poor path to the ant's search for new path, the parameters ρ in this paper is adjusted adaptively, when the optimal solution is not significantly improved in the sub-cycle, the values are adjusted according to the following formula:

$$\rho_n = \begin{cases} 0.90\rho_{n-1}, & 0.90\rho_{n-1} \geq \rho_{min} \\ \rho_{min}, & otherwise \end{cases} \tag{15}$$

In formula (15), to ensure that the opportunity for selecting nodes can be found under smaller circumstances, ρ_{min} is the minimum defined in the ant search process.

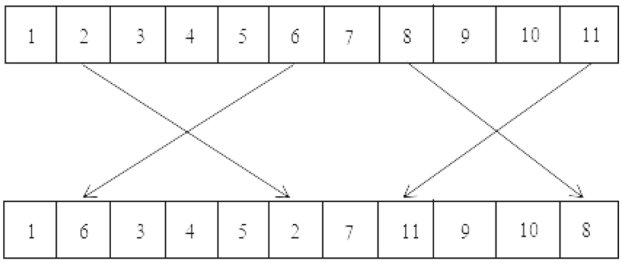


Fig. 1 Swap operator

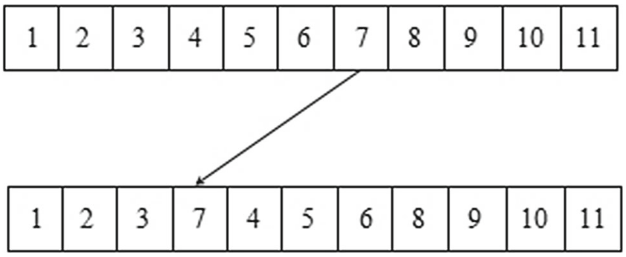


Fig. 2 Insert operator

Introduce mutation operation

In this paper, we introduce two kinds of mutation operators in ACO: swap operator and insert operator. The introduction to mutation operators not only increases the diversity of optimal solutions, but also avoids the convergence of the algorithm to local optima. We first briefly describe the mutation operator used in this paper.

Swap operator: The random exchange of gene positions within a gene sequence. In Fig. 1, the positions of genes 2 and 6, 8 and 11 are swapped.

Insert operator: The position in a gene sequence where a gene is randomly inserted into another gene. In Fig. 2, gene 7 was removed and inserted into the position of gene 4.

An example solution of TWVRP is given in Fig. 3, which is transformed into gene sequence, as shown in Fig. 4a. The gene sequence 4(a) was swapped to obtain Fig. 4b. The gene sequence represented in Fig.4b was inserted to obtain Fig. 4c. Where 0 is inserted into the gene sequence to represent the path of solution formation that meets the constraints.

Steps of the solution

The steps of solving TWVRP model with HACO are as follows:

Step 1: Initialization of parameters, including the maximum number of iterative NC, the number of ants V, the limited capacity Cap_k; the demand of node d_i for all $i \in \{1, 2, \ldots, n\}$.

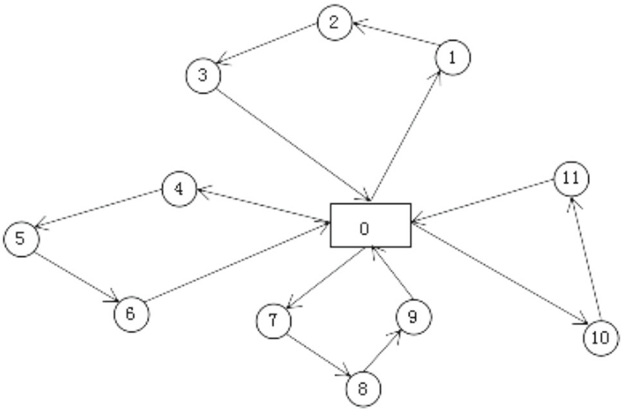
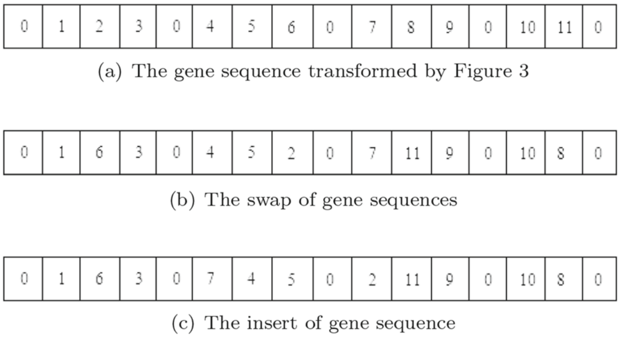


Fig. 3 A sample solution for TWVRP



0	1	2	3	0	4	5	6	0	7	8	9	0	10	11	0

(a) The gene sequence transformed by Figure 3

0	1	6	3	0	4	5	2	0	7	11	9	0	10	8	0

(b) The swap of gene sequences

0	1	6	3	0	7	4	5	0	2	11	9	0	10	8	0

(c) The insert of gene sequence

Fig. 4 Sequence codes

Step 2: Put ant $l = 1, 2, \ldots, v$ into the depot, and create an optional table $Cand_l = \{1, 2, \ldots, n\}$ to record all not visited nodes. After the ants pass through the nodes, they should record the nodes in the corresponding path record table $tabu_l = \emptyset$.

Step 3: Find the not visited node in the table $Cand_l = \{1, 2, \ldots, n\}$, and select the next node j to be accessed according to formula (11).

Step 4: Whether the ant l meets the constraints of vehicle capacity, time and node demand at the node j, if it meets the constraints, go to Step 5; otherwise, go to step 3.

Step 5: Put the selected node j into the table $tabu_l$,save the current result, and update the table $tabu_l$.

Step 6: Whether the ant l has traversed all the nodes, if so, go to Step 7; otherwise, go to step 2.

Step 7: Calculate the length of the path traveled by the ant, and conduct mutation operation on the path found this time. If the path after mutation operation meets the constraint conditions, record the optimal path and update the path records table.

Step 8: Update the pheromone according to Formula (12);

Step 9: To search the solution space comprehensively, parameter ρ needs to be adjusted adaptively according to formula (15);

Step 10: Is the current cycle number greater than the maximum iteration number. If so, the cycle will stop and output the result; if not, go to step 2.

Numerical analysis

To verify the performance of the proposed HACO, we apply it to Solomon's TWVRP standard instances, which has 56 data sets, each data set contains 100 data, and the data can be obtained from Solomon website.[1]

Solomon standard instances can be divided into six categories: C1, C2, R1, R2, RC1, and RC2. It can be divided into three types based on the geographic distribution of the customers, and the first type is the set C1 and C2, because the instance customers in these two sets are geographically concentrated; the second type is the set R1 and R2, because the instance customers in these two sets are randomly distributed in geographical locations, which are relatively discrete; the third type is the set RC1 and RC2, which are between two classes, both concentration and dispersal. Solomon standard instances can be further divided into two categories according to vehicle capacity fleet size and time window. The first category is C1, R1, and RC1, and these three sets have a smaller vehicle capacity, a smaller fleet size, and a narrower time window, so there are fewer customers served by vehicle; the second category is C2, R2, and RC2, and these three sets are opposite to the three sets in the first category, so there are more customers served by vehicle. In the numerical experiment, the standard instances are classified according to the second method above and divided into two categories to test and evaluate the algorithm.

Parameter setting

When the ACO is looking for the solution of the TWVRP, the parameters in the algorithm can affect the quality of the solution. There are usually two methods to select parameter values [41]. The first method is to set the parameters before the algorithm running, so that the parameters will not change into the running process. The second method is to set parameters during the running of the algorithm, but an initial value needs to be set before the running, and such parameters are constantly changing during the running process. Three sets of instances C101, R101, and RC01 are selected to analyze the parameters. When analyzing one of the parameters, the other parameters remained unchanged. The following numerical analysis using HACO and studying the parameters based on $g_k = 500$, $h_k = 1000$. By analyzing the optimal solution, the influence of parameters on the solution quality is explained, then the influence of parameters on the overall performance of the algorithm is explained.

1. Parameter α
   It can be seen from Fig. 5 that the approximate optimal objective values obtained using different parameters α. When the value of parameter α is around 1, we get a good value. Therefore, in the experiment, the best value we can get for the parameter α is equal to 1.
2. Parameter β
   As can be seen from Fig. 6, when the value of parameter β is between the interval [3,5], we get relatively good values. Therefore, the range of values for parameter β is [3,5].
3. Parameter γ
   As can be seen from Fig. 7, when the value of parameter γ is between the interval [2,4], the approximate optimal objective value obtained is relatively good. Therefore, the range of values for parameter γ is [2,4].
4. Parameter q_0
   As can be seen from Fig. 8, when the value of parameter q_0 is between the interval [0.3, 0.6], we get a better approximate optimal objective value. Therefore, the range of values for parameter q_0 is [0.3, 0.6].
5. Parameter ρ
   As can be seen from Fig. 9, when the value of parameter ρ is between the interval [0.3, 0.7], we get a relatively good approximate optimal objective value. The smaller are the value of ρ, the more pheromones are left on the path, so it is not easy to distinguish the better path, and the optimal solution is difficult to get. However, when ρ is large, because there are fewer pheromones on the poor path, the algorithm may converge on the local optimal. Therefore, the range of values for parameter ρ is [0.3, 0.7].

Experimental results

To illustrate the advantages and disadvantages of the proposed HACO, we analyze the experimental results from two aspects: first, the results obtained by HACO are compared with ACO; second, the results obtained by HACO are compared with other intelligent algorithms.

Comparison with the ACO

The Solomon standard instances are used as a test function to test the performance of HACO and evaluate the optimal solution quality. Tables 1 and 2 show the results for ACO and HACO.

[1] http://w.cba.neu.edu/solomon/problems.htm.

Fig. 5 Approximate optimal values under different α

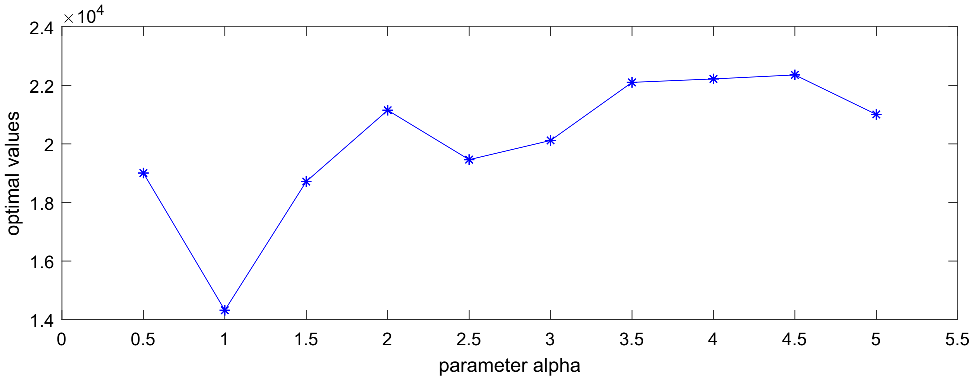


Fig. 6 Approximate optimal values under different β

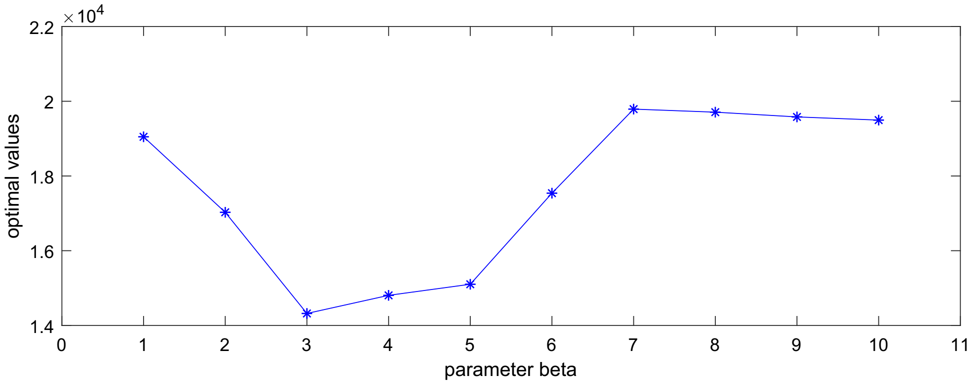


Fig. 7 Approximate optimal values under different γ

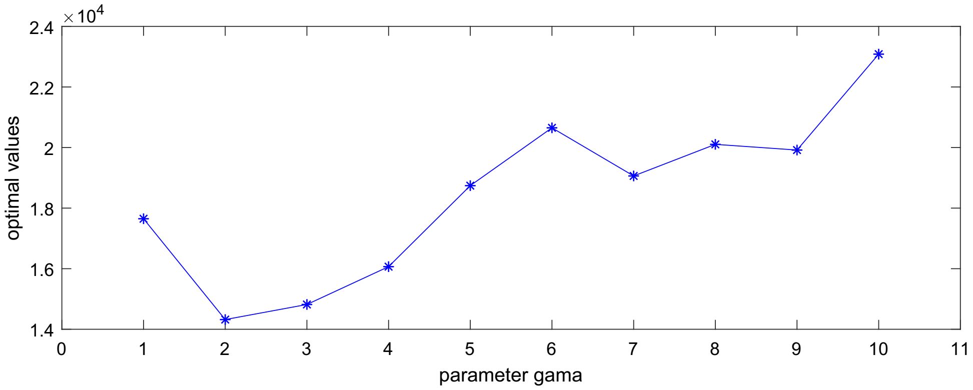


Fig. 8 Approximate optimal values under different $q0$

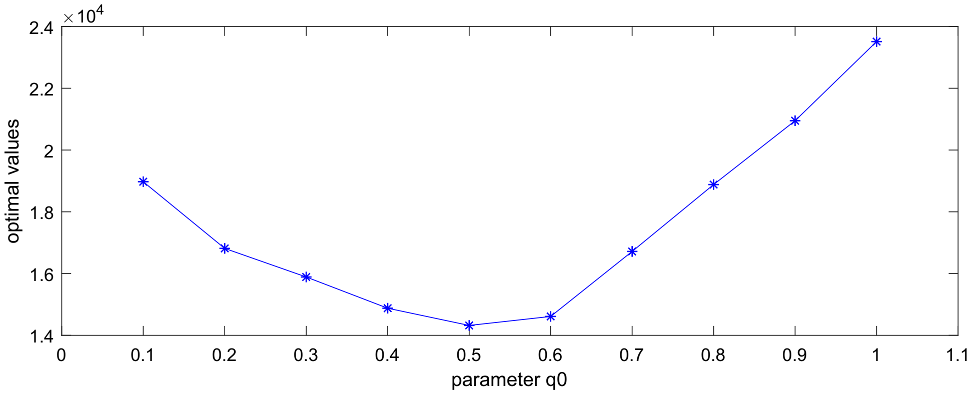


Fig. 9 Approximate optimal values under different ρ

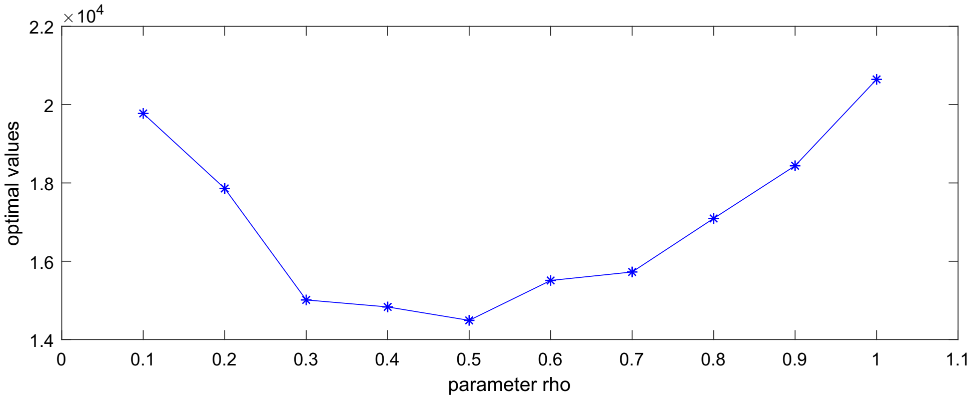


Table 1 Comparison with the ACO for the category of type 1 (C1, R1, and RC1)

Instance	ACO			HACO				
	Best result in 10 runs			Best result in 10 runs				
	Veh	Dis	$Cost$	Veh	Dis	$Gap_{Dis}(\%)$	$Cost$	$Gap_{Cost}(\%)$
C101	13	1299.02	19,798	13	**1262.53**	**−2.89**	**14320**	**−38.25**
C102	13	1538.73	19,673	13	1693.11	9.12	14627	−34.50
C103	11	1669.38	15093	11	**1530.39**	**−9.08**	**12549**	**−20.27**
C104	10	1278.60	12280	10	1307.09	2.18	11382	−7.89
C105	12	1271.48	16651	11	**1244.97**	**−2.13**	**12287**	**−35.52**
C106	13	1522.94	17800	13	**1460.96**	**−4.24**	**14496**	**−22.79**
C107	13	1367.31	17781	13	1377.25	0.72	13521	−31.51
C108	12	1399.51	14568	12	**1309.69**	**−6.86**	**13447**	**−8.34**
C109	11	1275.45	15593	11	**1199.90**	**−6.30**	**13671**	**−14.06**
R101	26	2588.94	34802	26	**2550.81**	**−1.49**	**34008**	**−2.33**
R102	23	2370.23	29570	23	**2343.94**	**−1.12**	**28536**	**−3.62**
R103	17	1973.61	21220	16	**1848.63**	**−6.76**	**21064**	**−0.74**
R104	13	1404.67	15610	13	1398.59	−0.43	16560	5.74
R105	19	1966.78	24318	19	1870.37	−5.15	25316	3.94
R106	16	1859.20	22026	16	**1843.72**	**−0.84**	**21038**	**−4.70**
R107	13	1647.18	16676	13	**1630.95**	**−1.00**	**15791**	**−5.60**
R108	12	1317.30	14459	12	**1285.82**	**−2.45**	**14447**	**−0.08**
R109	15	1760.21	18879	15	**1696.26**	**−3.77**	**18705**	**−0.93**
R110	14	1575.43	16707	14	**1566.01**	**−0.60**	**15615**	**−6.99**
R111	14	1512.73	16716	14	**1510.99**	**−0.12**	**16656**	**−0.36**
R112	12	1324.90	14433	12	**1272.96**	**−4.08**	**13390**	**−7.79**
RC101	21	2444.21	25654	21	**2350.94**	**−3.97**	**23750**	**−8.02**
RC102	16	2142.05	21450	16	**2132.71**	**−0.44**	**21379**	**−0.33**
RC103	13	1802.48	18049	13	**1791.28**	**−0.63**	**17074**	**−5.71**
RC104	12	1655.40	15794	12	**1643.07**	**−0.75**	**14833**	**−6.48**
RC105	19	2571.79	24691	19	**2362.44**	**−8.86**	**23636**	**−4.46**
RC106	15	1917.75	20196	15	1935.82	0.93	20221	0.12
RC107	14	1752.82	17028	14	**1733.07**	**−1.14**	**16913**	**−0.68**
RC108	13	1589.41	15766	13	**1567.73**	**−1.38**	**15677**	**−0.57**

In Tables 1 and 2, ACO and HACO run ten times and choose the best result. Where Veh represents the number of vehicles; Dis is the distance traveled by vehicle; Cost represents the cost of the vehicle; $Gap_{Dis}(\%)$represents the difference between the distance obtained by HACO and the distance obtained by ACO, and the expression is

Table 2 Comparison with the ACO for the category of type 2 (C2, R2, and RC2)

Instance	ACO			HACO				
	Best result in 10 runs			Best result in 10 runs				
	Veh	Dis	$Cost$	Veh	Dis	$Gap_{Dis}(\%)$	$Cost$	$Gap_{Cost}(\%)$
C201	3	591.56	3603.92	3	591.56	0	3603.92	0
C202	4	982.80	5270.53	4	**905.92**	**−8.49**	**5202.31**	**−1.31**
C203	3	978.60	5152.84	4	876.94	−11.59	5302.75	2.83
C204	3	994.06	5171.55	3	**979.53**	**−1.48**	**5155.46**	**−0.31**
C205	4	674.47	4800.37	4	**669.25**	**−0.78**	**4688.43**	**−2.39**
C206	4	776.75	4851.46	4	**753.34**	**−3.11**	**3996.40**	**−21.40**
C207	4	756.17	5065.52	4	**715.13**	**−5.74**	**4985.96**	**−1.60**
C208	4	773.63	4896.77	4	**720.31**	**−7.40**	**4785.73**	**−2.32**
R201	4	2063.83	8151.80	4	**2018.20**	**−2.26**	**8100.00**	**−0.64**
R202	4	2097.53	8376.00	4	**1977.10**	**−6.09**	**7862.00**	**−6.54**
R203	3	1665.79	5827.70	3	1761.75	5.45	5846.80	0.33
R204	3	1238.69	4444.20	3	**1195.32**	**−3.63**	**4371.20**	**−1.67**
R205	4	1495.27	5746.40	4	**1470.54**	**−1.68**	**5717.00**	**−0.51**
R206	3	1470.71	5487.50	3	**1455.13**	**−1.07**	**4693.30**	**−16.92**
R207	3	1339.77	4491.90	3	**1333.29**	**−0.49**	**4489.50**	**−0.05**
R208	3	1088.24	4226.10	3	**1044.05**	**−4.23**	**4222.50**	**−0.09**
R209	3	1501.85	5607.40	3	**1440.30**	**−4.27**	**5602.70**	**−0.08**
R210	3	1508.14	5627.80	3	1561.41	3.41	5670.90	0.76
R211	3	1084.07	4346.30	3	1189.92	8.90	4195.10	−3.60
RC201	5	2364.20	8444.56	5	**2298.07**	**−2.88**	**8360.24**	**−1.01**
RC202	4	2029.48	7220.93	4	2046.15	0.81	7467.31	3.30
RC203	4	1691.10	5987.02	4	**1684.92**	**−0.37**	**5764.29**	**−3.86**
RC204	3	1267.54	4396.78	3	**1245.31**	**−1.79**	**4378.33**	**−0.42**
RC205	5	2138.76	7304.04	5	**2096.55**	**−2.01**	**7233.18**	**−0.98**
RC206	4	1729.86	5729.86	4	1748.73	1.08	5814.34	1.45
RC207	4	1656.32	5694.47	4	1618.84	−2.32	5740.28	0.80
RC208	3	1357.87	4597.86	3	**1306.25**	**−3.95**	**4431.72**	**−3.75**

$$Gap_{Dis}(\%) = \frac{HACO_{Dis} - ACO_{Dis}}{HACO_{Dis}} \times 100(\%). \qquad (16)$$

$Gap_{Cost}(\%)$ represents the difference between the cost of HACO and the cost of ACO, and the expression is

$$Gap_{Cost}(\%) = \frac{HACO_{Cost} - ACO_{Cost}}{HACO_{Cost}} \times 100(\%). \qquad (17)$$

As shown in Table 1, when the number of vehicles is the same, the HACO results is better than ACO results, which shorten the distance and reduce the cost. However, there are six instances in Table 1 with poor effect, in which instance C102, C104, and C107 increase the driving distance while reducing the cost; instance R104 and R105 increase costs while reducing distances; RC106 not only increases the driving distance but also increases the cost. Except for the six instances that did not work well, the other instances did significantly better.

It can also be seen from Table 2 that compared with ACO, HACO has a better effect in most instances, which not only reduces the driving distance but also reduces the cost. Similarly, there are seven instances with poor effect in Table 7, among which the driving distance of instances R203, R210, RC202, and RC206 is increased and the cost is also increased; instance C203 and RC207 increase the cost while shortening the distance;instance R211 reduces cost but increases distance.

Comparisons between the other intelligent algorithms (node numbers are the same)

Each instance of Solomon's standard instance has 100 customers that are geographically distributed differently. Tables 3 and 4 compare HACO with other intelligent algorithms for the same number of customers. The boldface

Table 3 Comparison results of type 1 (C1, R1, and RC1)

Instance	SA			GA			HACO		
	Veh	Dis	Cost	Veh	Dis	Cost	Veh	Dis	Cost
C101	15	2135.20	25210.00	14	1901.83	21140.00	13	**1262.53**	**14320.00**
C102	13	1812.56	21369.00	13	1789.27	19953.00	13	**1693.11**	**14627.00**
C103	11	1901.99	19219.00	11	1666.86	16628.00	11	**1530.39**	**12549.00**
C104	10	1639.66	16596.00	10	1335.73	12578.00	10	**1307.09**	**11382.00**
C105	11	2161.45	23301.00	11	1311.57	13187.00	11	**1244.97**	**12287.00**
C106	13	1997.92	19630.00	13	1481.56	15022.00	13	**1460.96**	**14496.00**
C107	11	1182.75	12384.00	11	1052.57	12216.00	13	1377.25	13521.00
C108	12	1742.48	17873.00	12	1312.65	13597.00	12	**1309.69**	**13447.00**
C109	11	1718.28	17199.00	11	1214.12	14391.00	11	**1199.90**	**13671.00**
R101	20	2082.06	32544.00	22	1921.79	30201.00	26	2550.81	34008.00
R102	21	1989.32	26675.00	20	1711.83	26046.00	23	2343.94	28536.00
R103	15	1680.94	17273.00	18	1518.13	16734.00	16	1848.63	21064.00
R104	13	1537.25	18006.00	13	1410.95	17425.00	13	**1398.59**	**16560.00**
R105	17	1779.36	20360.00	16	1510.71	21858.00	19	1870.37	25316.00
R106	15	1764.23	18175.00	15	1468.54	17951.00	16	1843.72	21038.00
R107	13	1712.83	18390.00	13	1766.38	19241.00	13	**1630.95**	**15791.00**
R108	12	1447.08	14514.00	12	1309.60	14966.00	12	**1285.82**	**14447.00**
R109	13	1666.98	17247.00	14	1294.83	21040.00	15	1696.26	18705.00
R110	14	1645.49	16490.00	14	1573.89	16710.00	14	**1566.01**	**15615.00**
R111	13	1504.64	15646.00	13	1394.25	16043.00	14	1510.99	16656.00
R112	12	1375.96	13759.00	12	1317.88	14156.00	12	**1272.96**	**13390.00**
RC101	17	2322.98	22840.00	16	1738.53	18180.00	21	2350.94	23750.00
RC102	16	2173.43	26259.00	16	2389.70	25786.00	16	**2132.71**	**21379.00**
RC103	13	1854.61	20733.00	13	1856.08	20253.00	13	**1791.28**	**17074.00**
RC104	14	1672.42	17354.00	12	1702.10	16569.00	12	**1643.07**	**14833.00**
RC105	18	2134.18	21365.00	17	1783.33	22928.00	19	2362.44	23636.00
RC106	14	1846.32	25178.00	15	1635.40	21140.00	15	1935.82	20221.00
RC107	14	1788.89	17953.00	14	1762.27	18133.00	14	**1733.07**	**16913.00**
RC108	13	1678.36	17042.00	13	1574.27	16503.00	13	**1567.73**	**15677.00**

indicates that HACO gets better results than the comparison algorithm.

Tabu search algorithm and genetic algorithm are used as comparison algorithms in Tables 3 and 4. The Veh, Dis, and Cost are consistent with above meanings. Table 3 shows that when the number of customers is the same, the HACO has 62.07% better results than the comparison algorithm. Among them, when customers are concentrated on geography (set C1), the HACO has 88.89% better results than the comparison algorithm; when the customer distribution in the geographical location is dispersed (set R1), the HACO only has 41.67% better results, and the HACO is not very good. The HACO is also feasible when customers are geographically dispersed and concentrated, with 62.50% better results.

Similarly, Table 4 shows that when the number of customers is the same, the HACO has 60.71% better results than the comparison algorithm. Among them, the HACO has 75% better results than the comparison algorithm when customers are relatively concentrated (set C2); when customers are scattered (set R2), the HACO has only 54.55% better results, and the result obtained by the comparison algorithm is better. When customers are concentrated and dispersed, the HACO has 62.50% better results than the comparison algorithm.

Comparisons between the other intelligent algorithms (customer needs are the same)

In standard instances of Solomon, each instance has 100 customers with different needs, some with more and some with less. We classify the same or similar customer requirements in each instance, and the performance and convergence of the proposed algorithm are evaluated from the perspective of customer demand. To fully analyze the performance of

Table 4 Comparison results of type 2 (C2, R2, and RC2)

Instance	SA			GA			HACO		
	Veh	Dis	Cost	Veh	Dis	Cost	Veh	Dis	Cost
C201	3	1395.73	4212.00	3	888.34	4382.00	3	**591.56**	**3603.92**
C202	3	1580.84	6150.00	4	958.11	5483.00	4	**905.92**	**5202.31**
C203	4	1294.96	4967.00	4	965.19	3684.00	4	876.94	5302.75
C204	3	1282.14	5947.00	3	1028.03	5851.00	3	**979.53**	**5155.46**
C205	4	1127.15	6029.00	3	705.49	4994.00	4	**669.25**	**4688.43**
C206	3	1341.12	4702.00	3	760.77	4087.00	3	**753.34**	**3996.40**
C207	3	1462.22	5175.00	4	1028.50	5092.00	4	**715.31**	**4985.96**
C208	4	1372.35	4576.00	4	741.75	2053.00	4	720.31	4785.73
R201	4	2045.47	8873.00	4	1218.40	8819.00	4	2018.20	8100.00
R202	4	1238.68	2940.00	4	1151.54	3429.00	4	1977.10	7862.00
R203	3	1578.86	5788.00	6	881.41	5158.00	3	1761.75	5846.80
R204	3	1267.85	4714.00	5	1218.74	4484.00	3	**1195.32**	**4371.00**
R205	4	1542.28	7088.00	7	1475.60	6495.00	4	**1470.54**	**5717.00**
R206	3	1455.18	5417.00	4	1509.65	5150.00	3	**1455.13**	**4693.30**
R207	3	1344.84	5156.00	5	1348.00	5630.00	3	**1333.29**	**4489.50**
R208	3	1292.03	3278.00	4	1186.22	4043.00	3	1044.05	4222.50
R209	3	1513.12	6739.00	6	1475.50	6403.00	3	**1440.30**	**5602.70**
R210	3	1451.76	5863.00	6	941.74	5272.00	3	1561.41	5670.90
R211	4	1249.35	4356.00	6	1200.82	4709.00	3	**1189.92**	**4195.10**
RC201	6	2415.57	9047.00	7	2374.40	8769.00	5	**2298.07**	**8360.24**
RC202	4	21740.97	6145.00	7	1136.82	5284.00	4	2046.15	7467.31
RC203	4	1705.41	6812.00	5	1715.39	6095.00	4	**1684.92**	**5764.29**
RC204	3	1336.28	4417.00	5	1248.27	4645.00	3	**1245.31**	**4378.33**
RC205	4	1625.37	5250.00	8	1374.65	5224.00	5	2096.55	7233.18
RC206	4	1689.13	5523.00	6	1245.04	5442.00	4	1748.73	5814.34
RC207	4	1735.33	6902.00	7	1803.47	6918.00	4	**1618.64**	**5740.28**
RC208	4	1373.24	4784.00	5	1407.83	5031.00	3	**1306.25**	**4431.72**

Table 5 Comparison between the HACO and other intelligent algorithms for small instance C101 when customer requirements are the same

Demand	N	HACO				Reference [35]				Reference [42]			
		Veh	Dis	Cost	Time	Veh	Dis	Cost	Time	Veh	Dis	Cost	Time
10	52	8	**1307.76**	**11526.00**	142.48	8	1322.22	11566.00	165.01	8	1373.71	11540.00	156.70
20	26	5	939.16	6942.64	**48.89**	5	939.16	6942.64	52.82	5	967.71	7025.98	52.75
30	13	4	537.29	4575.96	**18.78**	4	537.29	4575.96	20.04	4	537.29	4575.96	20.35
40	7	2	**289.93**	**2308.37**	**9.53**	2	308.27	2349.06	10.70	2	308.27	2349.06	10.92
50	2	1	39.82	1039.87	**2.24**	1	39.82	1039.87	2.74	1	39.82	1039.87	2.59

Demand	*N*	Reference. [18]				GA				SA			
		Veh	Dis	Cost	Time	Veh	Dis	Cost	Time	Veh	Dis	Cost	Time
10	52	8	1328.31	11561.00	154.97	9	2212.29	11931.00	87.60	8	1352.08	11902.00	143.89
20	26	5	939.16	6942.64	54.33	6	967.71	7025.89	91.81	5	943.91	6993.66	49.83
30	13	4	537.29	4575.96	20.90	4	621.66	4634.73	76.63	5	462.38	4155.65	60.02
40	7	2	308.27	2349.06	10.76	2	329.37	3353.68	49.58	2	291.393	2518.37	28.53
50	2	1	39.82	1039.87	2.62	1	39.82	1039.87	17.25	1	39.82	1039.87	21.70

Table 6 Comparison between the HACO and other intelligent algorithms for small instance R101 with the same customer requirements

Demand	N	HACO				Reference [35]				Reference [42]			
		Veh	Dis	Cost	Time	Veh	Dis	Cost	Time	Veh	Dis	Cost	Time
$0 \le D < 10$	36	10	1108.06	13196.00	84.97	10	1045.79	13155.00	86.41	10	989.62	12112.00	87.14
$10 \le D < 20$	39	11	**1066.24**	**14120.00**	**94.53**	11	1084.99	15136.00	96.60	12	1103.99	15145.00	95.61
$20 \le D < 30$	19	7	**612.13**	**7616.43**	**35.04**	7	616.11	7630.85	36.82	7	666.87	8678.45	36.50
$30 \le D < 40$	5	3	242.55	3242.68	**6.43**	3	242.55	3242.68	6.87	3	242.55	3242.68	6.87
$D \ge 40$	1	1	46.04	1046.07	**0.92**	1	46.04	1046.07	1.02	1	46.04	1046.07	1.00

Demand	N	Reference [18]				**GA**				SA			
		Veh	Dis	Cost	Time	Veh	Dis	Cost	Time	Veh	Dis	Cost	Time
$0 \le D < 10$	36	10	1052.42	13183.00	88.63	13	1262.42	15177.00	110.39	5	682.18	12291.00	32.17
$10 \le D < 20$	39	11	1072.22	14188.00	98.46	11	1144.93	14175.00	95.70	11	1086.08	14740.00	95.32
$20 \le D < 30$	19	7	616.11	7630.85	35.28	7	619.16	7646.65	73.19	7	615.91	7622.27	37.30
$30 \le D < 40$	5	3	242.55	3242.68	6.82	3	303.65	4324.81	58.36	3	242.55	3242.68	27.35
$D \ge 40$	1	1	46.04	1046.07	0.96	1	46.04	1046.07	18.72	1	46.04	1046.07	16.27

Table 7 Comparison between the HACO and other intelligent algorithms for small instance RC101 when customer requirements are the same

Demand	N	HACO				Reference [35]				Reference [42]			
		Veh	Dis	Cost	Time	Veh	Dis	Cost	Time	Veh	Dis	Cost	Time
$0 \le D < 10$	17	4	548.83	5566.32	30.12	4	548.83	5566.32	29.51	4	546.73	5564.83	32.92
$10 \le D < 20$	41	9	**1115.08**	**12460.00**	**101.43**	9	1143.91	12475.00	104.27	9	1303.95	13410.00	103.68
$20 \le D < 30$	26	7	**926.77**	**8930.23**	**51.84**	7	936.50	9115.98	53.00	8	938.41	9124.66	56.38
$30 \le D < 40$	11	5	533.96	6591.03	**17.03**	5	533.96	6591.03	18.01	5	566.58	6611.62	18.12
$D \ge 40$	5	3	271.83	3271.85	**6.21**	3	271.83	3271.85	7.15	3	271.83	3271.85	6.72

Demand	N	Reference [18]				GA				SA			
		Veh	Dis	Cost	Time	Veh	Dis	Cost	Time	Veh	Dis	Cost	Time
$0 \le D < 10$	17	4	539.15	5562.93	30.53	4	434.49	5514.32	101.07	5	420.66	4344.89	30.26
$10 \le D < 20$	41	9	1143.91	12475.00	102.20	9	1301.15	13427.00	105.86	9	1149.86	12619.00	121.38
$20 \le D < 30$	26	7	936.50	9115.98	55.26	7	945.63	9376.23	72.29	7	937.18	9110.57	53.05
$30 \le D < 40$	11	5	533.96	6591.03	17.86	5	612.50	7691.09	59.73	5	533.96	6591.03	29.73
$D \ge 40$	5	3	271.83	3271.85	6.75	3	271.83	3271.85	18.36	3	271.83	3271.85	27.56

the proposed algorithm, we randomly selected six Solomon standard instances as test functions, and these six test functions are C101, R101, RC101, C201, R201, and RC101. To prove the effectiveness of mutation operation and adaptive strategy added to ant colony algorithm in this paper, we compared it with the improved ant colony algorithm in reference [18,35,42] as well as other algorithms GA, SA. Among them, reference [18] added local search method on the basis of ant colony algorithm, reference [35] proposed a new pheromone update formula, and reference [42] introduced adaptive strategy and exchange mechanism.

Tables 5, 6, 7, 8, 9, and 10 compare the HACO with the comparison algorithms in different sized instances with the same customer requirements. Where, Demand represents the demand of customers; N is the number of customers; Veh is the number of vehicles; Dis is the total distance traveled; Cost represents the total Cost of the vehicle; Time represents the running time of the algorithm. In addition, boldface indicates that HACO gets better results than the comparison algorithm.

Tables 5, 6, and 7 calculate the optimal results of different small instances (C101, R101, and RC101) with the proposed HACO and the comparison algorithm, respectively, and analyze the calculation results and the performance of the algorithm. Tables 5, 6, and 7 show that the HACO obtains at least 8 better results out of 15 comparisons when the number of vehicles used is the same or similar, which not only shorten the driving distance but also reduce the total cost of vehicles. When the number of vehicles and the total cost

Table 8 Comparison between the HACO and other intelligent algorithms for medium instance C201 when customer requirements are the same

Demand	N	HACO				Reference [35]				Reference [42]			
		Veh	Dis	Cost	Time	Veh	Dis	Cost	Time	Veh	Dis	Cost	Time
10	52	2	1219.30	5284.26	1484.69	3	1288.72	5304.53	150.56	2	1219.44	5286.27	152.97
20	26	2	**977.33**	**3039.87**	**51.85**	3	1014.29	4135.72	53.67	2	1039.82	4296.27	54.91
30	13	1	658.58	2574.51	21.30	3	578.35	1614.26	22.49	1	658.58	2574.51	23.10
40	7	1	213.97	1214.06	10.04	2	206.63	1182.07	11.67	1	213.97	1214.06	10.88
50	2	1	39.82	1039.82	**2.01**	1	39.82	1039.82	2.38	1	39.82	1039.82	2.63

Demand	N	Reference [18]				GA				SA			
		Veh	Dis	Cost	Time	Veh	Dis	Cost	Time	Veh	Dis	Cost	Time
10	52	3	1234.61	5299.47	152.91	3	470.35	4133.76	186.64	3	652.46	2552.16	133.60
20	26	2	1010.32	4038.54	53.45	3	1091.43	4365.19	96.65	4	998.92	3416.61	54.18
30	13	1	658.58	2574.51	22.30	1	594.08	1658.22	69.71	2	361.04	1549.73	29.49
40	7	1	213.97	1214.06	10.85	1	206.63	1182.07	42.34	2	206.63	1182.07	27.98
50	2	1	39.82	1039.82	2.33	1	39.82	1039.82	18.66	1	39.82	1039.82	24.30

Table 9 Comparison between the HACO and other intelligent algorithms for large instance R201 when customer requirements are the same

Demand	N	HACO				Reference [35]				Reference [42]			
		Veh	Dis	Cost	Time	Veh	Dis	Cost	Time	Veh	Dis	Cost	Time
$0 \le D < 10$	36	2	**948.10**	**3027.11**	**91.55**	2	987.21	3047.96	94.88	2	1001.76	3068.24	94.04
$10 \le D < 20$	39	2	**904.87**	**3903.03**	100.43	2	943.22	3927.11	101.24	2	937.75	3921.33	101.15
$20 \le D < 30$	19	2	**499.01**	**2562.93**	**36.12**	2	502.17	2640.05	36.41	2	546.25	2816.44	36.48
$30 \le D < 40$	5	1	270.86	1270.92	**6.61**	1	270.86	1270.92	6.78	1	270.86	1270.92	36.48
$D \ge 40$	1	1	46.04	1046.03	1.00	1	46.04	1046.03	1.10	1	46.04	1046.03	1.03

Demand	N	Reference [18]				GA				SA			
		Veh	Dis	Cost	Time	Veh	Dis	Cost	Time	Veh	Dis	Cost	Time
$0 \le D < 10$	36	2	987.21	3047.96	92.25	6	966.24	3172.28	94.85	4	965.44	3121.19	92.71
$10 \le D < 20$	39	2	918.13	3911.52	101.12	4	1150.74	3949.41	797.88	4	987.91	3939.11	132.32
$20 \le D < 30$	19	2	544.34	2803.26	36.57	3	505.15	2685.01	79.07	3	543.92	2647.26	36.89
$30 \le D < 40$	5	1	270.86	1270.92	6.66	1	270.86	1270.92	52.38	1	270.86	1270.92	36.36
$D \ge 40$	1	1	46.04	1046.03	0.99	1	46.04	1046.03	19.53	1	46.04	1046.03	15.49

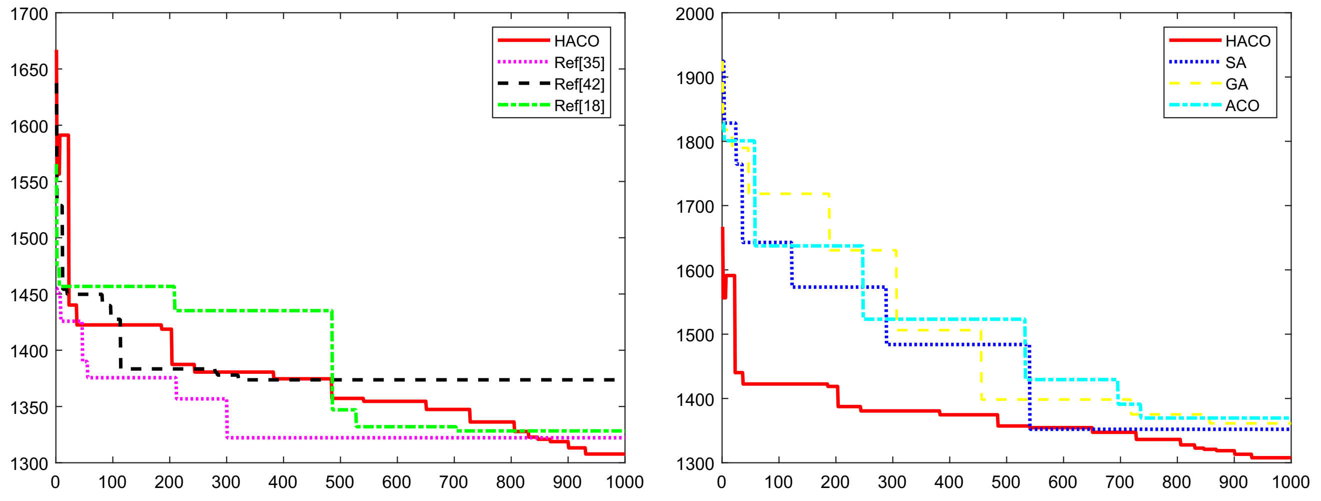
are the same, HACO takes less time to compute the optimal result than the comparison algorithm. In addition, these three tables also show that when solving small instances, no matter how customers are distributed among geography, the HACO results are better than the comparison algorithm.

Tables 8, 9, and 10 calculate the optimal results of medium instance (C201) and large instance (R201 and RC201), respectively, using the proposed HACO and the comparison algorithm, and further illustrate the performance of the proposed HACO by analyzing the results. As shown in Table 8, the proposed HACO does not get good results when solving medium instance, and only gets better results when customer demands are 20. However, HACO gets at least 8 better results out of 15 comparison results when solving the large-instance model. For the same number of vehicles, HACO gets better or similar results for total distance and total cost than the comparison algorithm, and the running time is also less than the other intelligent algorithms.

From Tables 5, 6, 7, 8, 9, and 10, it can be seen that when the customer demand is 10–30, except for medium instance C201, HACO gets better results for small and large instances than other intelligent algorithms. The hybrid algorithm proposed in reference [43] is also suitable for solving large instances problems, and it combines greedy algorithm with variable neighborhood search. Although the accuracy of the initial solution is improved, the convergence of the algorithm is not compared with other intelligent algorithms. The algorithm proposed in reference [44,45] can solve practical

Table 10 Comparison between the HACO and other intelligent algorithms for large instance RC201 when customer requirements are the same

Demand	N	HACO				Reference [35]				Reference [42]			
		Veh	Dis	Cost	Time	Veh	Dis	Cost	Time	Veh	Dis	Cost	Time
$0 \leq D < 10$	17	2	512.27	2517.23	28.31	2	517.21	2535.17	28.92	2	517.21	2535.17	28.76
$10 \leq D < 20$	41	3	**1183.36**	**4303.26**	**106.42**	3	1188.90	4323.45	107.49	3	1214.90	4374.58	107.93
$20 \leq D < 30$	26	2	**867.32**	**2997.92**	**55.34**	2	871.54	3091.32	55.42	2	891.93	3063.48	56.43
$30 \leq D < 40$	11	2	501.22	2530.98	**17.28**	2	501.22	2530.98	18.15	2	531.46	2616.03	18.11
$D \geq 40$	5	1	270.43	1270.48	**6.49**	1	270.43	1270.48	6.64	1	270.43	1270.48	6.77
Demand	N	Reference [18]				GA				SA			
		Veh	Dis	Cost	Time	Veh	Dis	Cost	Time	Veh	Dis	Cost	Time
$0 \leq D < 10$	17	2	517.21	2535.17	29.14	3	408.32	2311.93	98.98	4	375.48	1445.28	30.33
$10 \leq D < 20$	41	3	1214.90	4374.58	109.53	6	1238.52	4390.58	107.33	5	1218.88	4385.01	113.28
$20 \leq D < 30$	26	2	872.77	3024.85	57.10	3	896.95	3126.22	79.08	4	891.10	3123.62	59.31
$30 \leq D < 40$	11	2	501.22	2530.98	18.70	1	642.08	2633.74	18.75	3	507.79	2533.47	28.99
$D \geq 40$	5	1	270.43	1270.48	6.56	1	270.43	1270.48	23.16	1	270.43	1270.48	26.83



(a)

(b)

Fig. 10 The convergence curve of the optimal path

problems. However, on the basis of improving the solution quality, the convergence of the proposed hybrid algorithm in this paper is also analyzed, as shown in Fig. 10.

Figure 10 shows the convergence of the optimal path; subgraphs (a), (b), respectively, represent the convergence curves of different comparison algorithms when solving TWVRP.

(a) Comparison of HACO, Ref. [35], Ref. [42], and Ref. [18] in TWVRP.

(b) Comparison of HACO, SA, GA, and ACO in TWVRP.

Practical application

To test the effectiveness and practicability of the proposed HACO, we apply it to an actual example. Take the logistics of fresh fruits and vegetables in Xingqing District, Yinchuan, Ningxia as an example. We need to send fruits and vegetables from the warehouse (Xianfeng fruit and vegetable distribution center) to 13 distribution points (13 supermarkets in Xingqing District). Figure 11 shows the location of the warehouse (0) and 13 supermarkets on the map.[2] And the distance between each two point is shown in Table 11.

When delivering goods from the distribution center to each distribution point, the transport vehicle is composed of sev-

[2] https://map.baidu.com/@11815234,4627365,13z.

Table 11 Distance between any two distribution points and warehouses (m)

node	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	4319.97	0												
2	3600.32	3423.51	0											
3	2190.46	2389.60	6496.76	0										
4	636.51	751.54	4693.87	3630.99	0									
5	1401.14	1219.43	3199.73	2204.81	3398.16	0								
6	2255.93	3839.21	2492.60	3918.52	3207.87	3847.98	0							
7	266.15	110.92	4087.92	3336.01	2410.09	707.59	1135.30	0						
8	3053.16	3042.28	4807.83	5596.93	4285.24	3651.14	3774.77	14202.78	0					
9	2082.25	2257.03	6247.70	4936.15	716.72	1566.84	3066.44	18601.23	2244.09	0				
10	452.76	339.73	3978.96	3502.96	2588.19	1075.87	1317.07	16912.56	428.76	2723.34	0			
11	2962.62	3018.41	5579.24	6028.62	3671.38	3468.77	3998.29	14486.57	3129.07	992.84	4130.70	0		
12	2695.06	2666.65	4424.77	5155.24	4099.16	3309.50	3345.50	14573.60	2765.47	442.14	4384.20	2338.99	0	
13	4116.69	2473.58	5416.36	1841.93	1556.27	3788.51	3187.91	3144.79	4627.54	2526.65	2758.12	1209.91	1009.73	0

Table 12 Time window and demand for each distribution point

point	a_i	b_i	Demands(ton)	Service time(min)
0	9:40	11:40	0	0
1	10:10	10:25	15	20
2	10:00	10:15	16	15
3	9:50	10:05	15	15
4	10:20	10:45	10	13
5	11:00	11:25	5	8
6	10:50	11:15	7	9
7	10:35	10:55	5	8
8	10:00	10:30	25	17
9	10:10	10:35	22	19
10	10:45	11:10	26	21
11	10:05	10:40	18	11
12	10:25	10:50	13	14
13	10:20	10:45	16	12

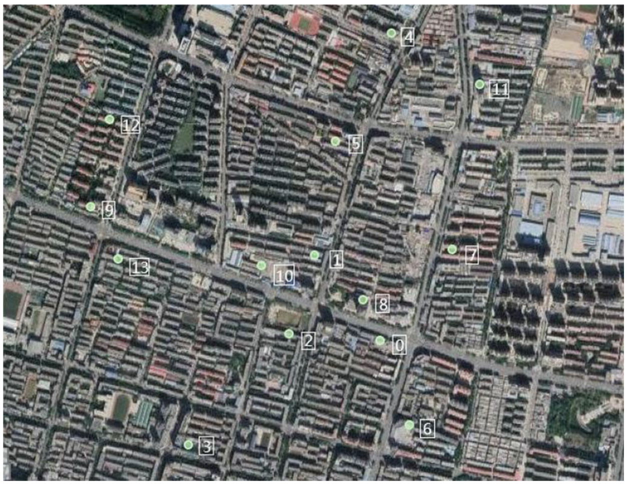


Fig. 11 Geographical location of distribution center and distribution point

eral trucks of the same type. The maximum carrying capacity of the truck is 40 tons, and the average speed is 60 km per hour. The fixed cost of transportation for each truck is 600 yuan per kilometer, and the transportation cost is 5 yuan per kilometer. Table 12 gives the specific situation of distribution centers and supermarkets according to the marking sequence in Fig. 11. It can be seen from Table 12 that supermarkets try to avoid the rush hours of traffic sections when purchasing goods, and the time window of supermarkets is almost between 9:40 and 11:40. Therefore, the study of this paper does not consider the traffic congestion, and the vehicle can deliver the goods within the time window specified by the supermarket.

The specific conditions of the actual problem can be known through Tables 11 and 12. The goal is to reduce the

Table 14 HACO results of solving TWVRP instance

Veh	Route	Distance (m)	Sum of demands (ton)
1	0–8–1–0	10415.41	40
2	0–11–9–0	6037.71	40
3	0–13–12–7–5–0	21808.75	39
4	0–3–2–6–0	13435.75	38
5	0–4–10–0	3677.46	36
Total		55375.08	193

Table 13 ACO results of solving TWVRP instance

Veh	Route	Distance (m)	Sum of demands (ton)
1	0–1–11–0	10301.00	33
2	0–13–12–7–0	19966.17	34
3	0–3–2–0	12287.54	31
4	0–9–4–0	3435.48	32
5	0–8–6–0	9083.86	32
6	0–10–5–0	2929.77	31
Total		58003.82	193

total cost of the distribution center by minimizing the number of vehicles and reducing the total driving distance of vehicles. Therefore, we apply ACO and HACO to solve the TWVRP instance. The practicability and effectiveness of the proposed HACO is analyzed and verified by comparing the results of the two algorithms, and the results are shown in Tables 13 and 14.

It can be seen from Table 13 that six vehicles are needed to solve the TWVRP instance using ACO, but these six vehicles are not fully utilized. According to the last line, the total distance is 58003.82 m and the fleet needs to spend 3890.02 yuan to complete all the deliveries. However, five vehicles are needed when HACO is used to solve the TWVRP example in Table 14, and these five vehicles are almost fully utilized compared with the six vehicles for Table 13. The total distance are 55375.08 m, which is 2628.74 m less than the total distance in Table 13. The cost of the fleet to complete all deliveries is 3276.88 yuan and decreases from 613.14 yuan. Therefore, through the application and analysis of examples, it can be shown that the proposed HACO is effective and has practical significance.

Conclusion

This paper mainly discusses the problem of vehicle routing with time window, to reduce the transportation distance and the transportation cost of vehicles, so a hybrid ant colony algorithm is proposed in this paper. Compared with the basic ant colony algorithm, the HACO changes the pheromone volatile factor from a fixed constant to a dynamic value, and this means that the optimal solution to the problem can be found by constantly adjusting the numerical values during the operation of the algorithm to improve the convergence of the algorithm. The pheromone updating formula for the ACO is improved, and with the participation of adaptive volatile factors, the pheromone concentration can increase or decrease according to the advantages and disadvantages of the path. To test the performance of HACO, we apply it to Solomon instances and the results showed that the proposed algorithm has good performance. The future research direction will consider the multi-objective green vehicle routing problem, not only to optimize the vehicle distance and cost, but also to take customer satisfaction on the service as the optimization goal. In addition, reducing vehicle exhausted emissions has been a new idea in the research of vehicle routing problems in recent years, so carbon dioxide emissions should also be taken as an optimization goal.

References

1. Zhang HZ, Zhang QW, Ma L, Zhang ZY, Liu Y (2019) A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows. Inf Sci 490:166–190
2. Ralphs TK, Kopman L, Pulleyblank WR, Trotter LE (2003) On the capacitated vehicle routing problem. Math Progr 94(2–3):343–359
3. Hernandez F, Feillet D, Giroudeau R, Naud O (2016) Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows. Eur J Oper Res 249(2):551–559
4. Schneider M (2016) The vehicle-routing problem with time windows and driver-specific times. Eur J Oper Res 250(1):101–119

5. Koc C, Bektas T, Jabali O, Laporte G (2015) A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows. Comput Oper Res 64:11–27
6. Bertsimas D, Simchi-Levi D (1993) A new generation of vehicle routing research: robust algorithms addressing uncertainty. Oper Res 44(2):286–304
7. Ombuki B, Ross BJ, Hanshar F (2006) Multi-objective genetic algorithms for vehicle routing problem with time windows. Appl Intell 24(1):17–30
8. Ghoseiri K, Ghannadpour SF (2010) Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. Appl Soft Comput 10(4):1096–1107
9. Wang KG, Gao YL (2019) Application of differential evolution algorithm based on mixed penalty function screening criterion in imbalanced data integration classification. Mathematics 7(12):1237
10. Wang KG, Gao YL (2019) Topology structure implied in β-Hilbert space, heisenberg uncertainty quantum characteristics and numerical simulation of the DE algorithm. Mathematics 7(4):330
11. Ho SC, Haugland D (2004) A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. Comput Oper Res 31(12):1947–1964
12. Belhaiza S, Hansen P, Laporte G (2014) A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. Comput Oper Res 52:269–281
13. Wang Z, Li Y, Hu XP (2014) A heuristic approach and a tabu search for the heterogeneous multi-type fleet vehicle routing problem with time windows and an incompatible loading constraint. Comput Ind Eng 89:162–176. https://doi.org/10.1016/j.cie.2014.11.004
14. Chiang WC, Russell RA (1996) Simulated annealing metaheuristics for the vehicle routing problem with time windows. Ann Oper Res 63(1):3–27
15. Deng AM, MAO C, Zhou YT (2009) Optimizing research of an improved simulated annealing algorithm to soft time windows vehicle routing problem with pick-up and delivery. Syst Eng Theory Pract 29(5):186–192
16. Wang C, Mu D, Zhao F, Sutherland JW (2015) A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup delivery and time windows. Comput Ind Eng 83:111–122
17. Yu B, Yang ZZ (2011) An ant colony optimization model: the period vehicle routing problem with time windows. Transp Res Part E: Log Transp Rev 47(2):166–181
18. Ding QL, Hu XP, Sun LJ, Wang YZ (2012) An improved ant colony optimization and its application to vehicle routing problem with time windows. Neurocomputing 98:101–107
19. Wang Y, Wang L, Chen GC, Cai ZQ, Zhou YQ, Xing LN (2020) An improved ant colony optimization algorithm to the periodic vehicle routing problem with time window and service choice. Swarm Evolut Comput. https://doi.org/10.1016/j.swevo.2020.100675
20. Geng N, Chen ZT, Nguyen QA, Gong DW (2021) Particle swarm optimization algorithm for the optimization of rescue task allocation with uncertain time constraints. Complex Intell Syst. https://doi.org/10.1007/s40747-020-00252-2
21. Ünal AN, Kayakutlu G (2020) Multi-objective particle swarm optimization with random immigrants. Complex Intell Syst 6(3):635–650
22. Qin SF, Sun CL, Zhang GC, He XJ, Tan Y (2020) A modified particle swarm optimization based on decomposition with different ideal points for many-objective optimization problems. Complex Intell Syst 6(2):263–274
23. Liu F, Zhang JW, Liu T (2020) A PSO-algorithm-based consensus model with the application to large-scale group decision-making. Complex Intell Syst 6(2):287–298
24. Ebrahimnejad A, Enayattabr M, Motameni H, Garg H (2021) Modified artificial bee colony algorithm for solving mixed interval-valued fuzzy shortest path problem. Complex Intell Syst. https://doi.org/10.1007/s40747-021-00278-0
25. Yang Y, Duan Z (2020) An effective co-evolutionary algorithm based on artificial bee colony and differential evolution for time series predicting optimization. Complex Intell Syst 6(2):299–308
26. Wang H, Wang WJ, Zhou XY, Zhao J, Wang Y, Xiao SY, Xu MY (2020) Artificial bee colony algorithm based on knowledge fusion. Complex Intell Syst. https://doi.org/10.1007/s40747-020-00171-2
27. Pandey AC, Tikkiwal VA (2021) Stance detection using improved whale optimization algorithm. Complex Intell Syst. https://doi.org/10.1007/s40747-021-00294-0
28. Enayattabar M, Ebrahimnejad A, Motameni H (2019) Dijkstra algorithm for shortest path problem under interval-valued Pythagorean fuzzy environment. Complex Intell Syst 5(2):93–100
29. Colorni A, Dorigo M, Maniezzo V (1991) Distributed optimization by ant colonies. In: Proceedings of ECAL91-European conference on artificial life
30. Bullnheimer B, Hartl RF, Strauss C (1997) Applying the ant system to the vehicle routing problem. In: International Conference on Metaheuristics 1–12
31. Yu B, Yang ZZ, Yao B (2009) An improved ant colony optimization for vehicle routing problem. Eur J Oper Res 196(1):171–176
32. Mavrovouniotis M, Yang S (2011) A memetic ant colony optimization algorithm for the dynamic travelling salesman problem. Soft Comput 15(7):1405–1425
33. Wang XY, Choi TM, Liu HK, Yue XH (2016) A novel hybrid ant colony optimization algorithm for emergency transportation problems during post-disaster scenarios. IEEE Trans Syst Man Cybern Syst 48(4):545–556
34. Jabir E, Panicker V, Sridharan R (2017) Design and development of a hybrid ant colony-variable neighbourhood search algorithm for a multi-depot green vehicle routing problem. Transp Res Part D Transp Environ 57:422–457
35. Li YB, Soleimani H, Zohal M (2019) An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives. J Clean Prod 227:1161–1172
36. Dorigo M (1992) Optimization learning and natural algorithms. Politecnico di Milano, Italy
37. EI-Sherbeny NA (2010) Vehicle routing with time windows: an overview of exact, heuristic and metaheuristic methods. J King Saud Univ Sci 22(3):123–131
38. Dorigo M, Maniezzo V, Colorni A (1996) The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part B 26(1):29–41
39. Dorigo M, Bonabeau E, Theraulaz G (2000) Ant algorithms and stigmergy. Fut Gen Comput Syst 16(8):851–871
40. Dorigo M, Blum C (2005) Ant colony optimization theory: a survey. Theor Comput Sci 344(2–3):243–278. https://doi.org/10.1016/j.tcs.2005.05.020
41. Ghannadpour SF, Noori S, Tavakkoli-Moghaddam R, Ghoseiri K (2014) A multi-objective dynamic vehicle routing problem with fuzzy time windows: model, solution and application. Appl Soft Comput 14:504–527
42. Huang SH, Huang YH, Blazquez CA, Paredes-Belmar G (2018) Application of the ant colony optimization in the resolution of the bridge inspection routing problem. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2018.01.034
43. Zhou YZ, Huang JC, Shi JM, Wang R, Huang KH (2021) The electric vehicle routing problem with partial recharge and vehicle recycling. Complex Intell Syst. https://doi.org/10.1007/s40747-021-00291-3
44. Feng GQ, Xu P, Cui DL, Dai XW, Liu H, Zhang Q (2021) Multi-stage timetable rescheduling for high-speed railways: a dynamic programming approach with adaptive state generation. Complex Intell Syst. https://doi.org/10.1007/s40747-021-00272-6

45. Jie J, Liu KR, Zheng H, Wang BB, Dai R (2021) High dimensional feature data reduction of multichannel sEMG for gesture recognition based on double phases PSO. Complex Intell Syst. https://doi.org/10.1007/s40747-020-00232-6

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.