

Research Article

A Hybrid Ant Colony Optimization Algorithm for Multi-Compartment Vehicle Routing Problem

Ning Guo ^{1,2}, Bin Qian ^{1,3}, Rong Hu,^{1,3} Huai P. Jin,^{1,3} and Feng H. Xiang¹

¹Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China

²Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology, Kunming 650500, China

³Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming 650500, China

Correspondence should be addressed to Bin Qian; bin.qian@vip.163.com

Received 20 August 2020; Revised 15 September 2020; Accepted 28 September 2020; Published 21 October 2020

Academic Editor: Qiang Chen

Copyright © 2020 Ning Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The multi-compartment vehicle routing problem (MCVRP) has been applied in fuel or food delivery, waste collection, and livestock transportation. Ant colony optimization algorithm (ACO) has been recognized as an efficient method to solve the VRP and its variants. In this paper, an improved hybrid ant colony optimization algorithm (IHACO) is proposed to minimize the total mileage of the MCVRP. First, a probabilistic model is designed to guide the algorithm search towards high-quality regions or solutions by considering both similar blocks of customers and customer permutations. Then, a heuristic rule is presented to generate initial individuals to initialize the probabilistic model, which can drive the search to the high-quality regions faster. Moreover, a new local search using the geometry optimization is developed to execute exploitation from the promising regions. Finally, two types of variable neighborhood descent (VND) techniques based on the speed-up search strategy and the first move strategy are devised to further enhance the local exploitation ability. Comparative numerical experiments with other algorithms and statistical analyses are carried out, and the results show that IHACO can achieve better solutions.

1. Introduction

The multi-compartment vehicle routing problem (MCVRP) is a variant of the well-known vehicle routing problem (VRP) [1]. Compared with the standard VRP, the MCVRP can simultaneously transport different types of products, which have to be placed in separate compartments. Since the customers probably have different demands for various products whose individual characteristics are so different that they cannot be mixed, each vehicle is divided into several compartments with a certain capacity for satisfying customers' needs. There are many industrial fields in which the multi-compartment vehicles are employed. The first example of the MCVRP arises in the process of supplying the fuels, where a lot of vehicles or ships with some tanks of various capacities are used to settle this problem [1–7]. Another example is the transportation of food, where different degrees of refrigeration goods are stored in different compartments in one vehicle [8–11]. In addition, the

MCVRP has also been widely applied in the waste collection, where separate compartments are required to convey different kinds of garbage [12–14]. Despite the practical applications, the solution of MCVRP has been rarely studied. Thus, this paper aims to minimize the total mileage for such a problem.

Since the VRP is an NP-hard problem and it can be reduced to the MCVRP, the latter issue is also NP-hard. Some operational research algorithms were presented to resolve small-scale problems for the MCVRP [1, 15–20]. However, with the increasing problem size, these algorithms are usually of limited use owing to their excessive time-consuming nature. Then, a variety of meta-heuristic algorithms were proposed to obtain desirable solutions within a reasonable calculation time [21–24]. There are many intelligent algorithms for solving nonlinear problems [25–27]. The ant colony optimization algorithm (ACO) is one of the highly efficient meta-heuristic or intelligent algorithms. The ACO is inspired by the intelligent behavior of ants in

searching for food, during which the ants leave the so-called pheromone on the road. The ants exchange information with other ants through the pheromone. The higher the number of pheromones on a path, the more likely the ants are to go after that path, and vice versa. In this way of the pheromone positive feedback mechanism, the ants can find the shortest path to the food location.

Because the ACO has a strong global exploration and distributed computing capability, it has been successfully applied to handle different kinds of optimization problems. While solving the VRP and the complex problems derived from it, ACO has been usually adopted by many researchers. The existing results usually use only pheromone concentration matrix (hereinafter referred to as pheromone matrix) to construct a probabilistic model to generate new individuals. The pheromone matrix learns and accumulates similar blocks of customers from every outstanding individual. Here any two successive visited customers in an individual compose one block. Mavrovouniotis and Yang [28] proposed an ACO with immigrant schemes for the dynamic vehicle routing problem. In this ACO, only the pheromone matrix was utilized to get information of each block, and its initial value was set to $(1/C^m)$, where (C^m) is the corresponding path length of a solution constructed by a nearest neighbor heuristic. Schyns [29] presented an ACO to settle the responsive dynamic vehicle routing problem with time window. The pheromone matrix was adopted merely and initialized to $(1/nL_0)$, where n is the number of customers and L_0 is the length of the route formed on the basis of the FIFO (first in first out) principle. Besides, a local search basic heuristic was developed to improve the algorithm performance. Wang et al. [30] designed a novel ACO called AMR to solve the vehicle routing problem. In AMR, only the pheromone matrix was used and its initial value was a small positive constant denoted by (ph_{\min}) . In addition, the saving algorithm was introduced to enhance the performance of AMR. Huang et al. [31] employed an ACO combined with a local search method in the resolution of the routing problem with bridge inspection tasks. Only the pheromone matrix was applied, and the initial pheromone was set to $(1/(N \cdot J_{\psi^{\text{ini}}}))$, where N is the total number of nodes and $(J_{\psi^{\text{ini}}})$ is the cost of the initial solution obtained by the first ant. Moreover, ACO has been rarely used to solve the MCVRP. Reed et al. [21] added a technique of clustering and local search of 2-opt to ACO in their study. Only the pheromone matrix was utilized and initialized to $(1/nL^{\text{neig}})$, where n is the customers' number and (L^{neig}) is the total length of the nearest neighbor path. Motivated by the work of Reed et al. [21], Abdulkader et al. [22] proposed a hybridized ant colony (HAC) algorithm with local search schemes. The pheromone matrix was adopted alone, and the initial value was set to $(1/L)$, where L is the length of a path which was generated randomly.

As it can be seen from above literatures, the probabilistic model only consists of the pheromone matrix in which only the relationship between two consecutive customers is considered. However, as mentioned earlier, the ACO generates new individuals based on the probabilistic model which stores the information of excellent individuals. Obviously, the more comprehensive the information of

excellent individuals are learned, the easier it is to guide the algorithm to reach different high-quality regions as soon as possible. Hence, only the information accumulation of similar blocks in the customer sequence is incomplete and limited. Besides, although pheromone matrices in different literatures above are initialized to different values, the values in the pheromone matrix of each literature are the same. This means that, at the beginning of the algorithm, there is no valuable information in the pheromone matrix. With the iteration of the algorithm, when the information in the pheromone matrix accumulates to a certain amount, it will guide new individuals towards high-quality areas. Invisibly, the efficiency of the algorithm is reduced in this way. Moreover, it is also found that the introduction of local search on the basis of the effective neighborhood operation can further obtain higher performance of the algorithm. Hence, in this research, an improved hybrid ant colony optimization algorithm (IHACO) is devised to solve the MCVRP. The salient features of this IHACO are as follows:

- (1) A new matrix is designed to save the whole order information of customers. Together with the pheromone matrix, the probabilistic model of IHACO is formed to record the worthy information of optimal individuals. Compared with the previous work, this design can learn and retain valuable information more comprehensively.
- (2) Several initial individuals produced by a heuristic rule are used to initialize the probabilistic model so that some valuable information can be utilized at the beginning of the algorithm. Then, the speed of reaching high-quality solution regions will be accelerated, and thus, the efficiency of the algorithm is raised.
- (3) Different from the majority of current local searches where 2-opt is commonly used within the same route, a new local search is proposed by adopting the geometry optimization method to perform narrow but productive exploitation.
- (4) In the IHACO's local search, three techniques are devised to enhance the exploitation capability from many aspects: two kinds of VND are used to increase the search depth of the algorithm, a speed-up search strategy based on the problem's properties is used to accelerate the evaluation process of individuals, and a first move strategy is used to enrich search regions.

The remainder of this paper is organized as follows: Section 2 introduces the mathematical model for the MCVRP; Section 3 describes the proposed IHACO for details; Section 4 presents computational experiments and analyses; and finally, Section 5 gives some conclusions and suggestions of future work.

2. Problem Description and Model Formulation

The MCVRP is defined on an undirected graph $(G = (N, A))$ with a set of vertices $N = \{0, 1, \dots, n\}$ and a set of arcs $A = \{(i, j): i, j \in N, i \neq j\}$. Denote d_{ij} as the distance

from i to j . Node 0 in N represents the depot, and $(N_c = (N/\{0\}))$ are nodes of customers. The depot has an available homogeneous fleet. These identical vehicles start from the depot, visit the customers distributed in the service area, and go back to the depot in the end. All vehicles are equipped with multiple compartments whose number equals to the number of products. Each customer ($i \in N_c$) has the quantity q_{ip} to be carried for the product p , and the compartment capacity reserved for this product does not exceed Q_p . Moreover, each customer is visited exactly once by only one vehicle.

To formulate the model, some sets, parameters, and variables adopted in the proposed problem are listed in Table 1.

Then, the integer linear program of the MCVRP can be described as follows:

$$\text{minimize} \cdot \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij}, \quad (1)$$

subject to

$$\sum_{j \in N} x_{ij} = 1, \quad \forall i \in N_c, \quad (2)$$

$$\sum_{i \in N} x_{ij} = 1, \quad \forall j \in N_c, \quad (3)$$

$$\sum_{i \in N_c} x_{0i} = \sum_{j \in N_c} x_{j0} = 1, \quad (4)$$

$$q_{ip} \leq Q_{ip} \leq Q_p, \quad \forall i \in N_c, p \in P, \quad (5)$$

$$Q_{ip} - Q_{jp} + Q_p x_{ij} \leq Q_p - q_{jp}, \quad \forall i \in N_c, p \in P, \quad (6)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in N, i \neq j. \quad (7)$$

Equation (1) is the objective function indicating the total mileage for the problem. Constraints (2) and (3) imply that one customer is visited just once by only one vehicle. Constraint (4) guarantees that each vehicle starts and ends its trip at the depot. Constraint (5) ensures that the total quantity delivered by each vehicle after serving customer i is less than its capacity for this product. Constraint (6) imposes the connectivity requirement of the feasible routes. Both inequalities (5) and (6) are subtour elimination constraints. Constraint (7) describes the domains of the decision variable.

3. IHACO for the MCVRP

This section describes the implementation of the proposed IHACO in detail, including the solution representation, establishment, and initialization of the probabilistic model, formation of the individuals, update of probability matrices, speed-up VND-based local search, and the complete IHACO process.

3.1. Solution Representation. Usually, the encoding rules of ACO for solving VRP are that each ant represents one vehicle, the path of each ant represents the driving path of each vehicle, the depot is represented by the number 0, and the customer is represented by a natural number [32]. In this paper, the decimal coding based on customer arrangement is still used, but the path of each ant which is denoted (π_{ant}) represents the path of all customers served by the depot, and the depot is a separator for different vehicles. For example, the path sequence of the ant i is $(\pi_{\text{ant}(i)}) = [0, 1, 5, 3, 7, 0, 6, 2, 8, 4, 0]$ which means that vehicle 1 leaves the depot 0, visits customers 1, 5, 3, and 7 in turn, and then returns to the depot 0; vehicle 2 leaves the depot 0, visits customers 6, 2, 8, and 4 in turn, and finally returns to the depot 0.

3.2. Probabilistic Model and Its Initialization. The performance of ACO has strong connections with the probabilistic model. A better choice of the model is critical to improve the efficiency of the algorithm. In this research, two probability matrices are employed to form the probabilistic model. Besides, a few good individuals produced by a heuristic method are used to initialize these two probability matrices. This initialization method makes the algorithm converge to the high-quality solutions faster.

3.2.1. Probabilistic Model. In the traditional ant colony algorithm, the probabilistic model usually refers to ants' pheromone matrix which is obtained by calculating the pheromone concentration (τ_{ij}) . This pheromone value represents the probability that the customer j appears immediately after the customer i . In other words, the pheromone matrix only reflects similar blocks of customers presented in the chosen individuals. But it is not enough to accumulate this kind of information alone. Counting the chance of customer j appearing before or in the current position is also important. Hence, a new probability matrix based on the order of customers called sequence information matrix is established. Then, the probabilistic model in this paper contains a pheromone matrix and a sequence information matrix.

In a subpath visited by one vehicle, let (s_{kj}) be the appearance times that customer j before or in position k in the chosen sequences. Next, the specific probability value in the sequence information matrix is determined by

$$\mu_{kj} = \frac{s_{kj}}{\sum_{l \in N_k} s_{k,l}}, \quad (8)$$

where N_k is the set of all available customers not already served by the current vehicle until position k . Here is an example to compute (μ_{kj}) . Suppose that the path taken by three ants are $\pi_{\text{ant}(1)} = [0, 1, 2, 3, 0, 4, 0]$, $\pi_{\text{ant}(2)} = [0, 2, 3, 4, 0, 1, 0]$, $\pi_{\text{ant}(3)} = [0, 3, 4, 0, 1, 2, 0]$, and $\pi_{\text{ant}(4)} = [0, 3, 2, 1, 4, 0]$, and then, s_{kj} is given as follows:

TABLE 1: Notations applied in the MCVRP model.

Symbol	Description of symbol
N	Node set ($N = 0 \cup N_c$)
N_c	Customer subset
P	Product set
d_{ij}	Distance between nodes i and j
q_{ip}	Quantity of product p to be carried from node i
Q_p	Capacity of the vehicle compartment dedicated to product p
x_{ij}	Equals 1 if the vehicle travels from node i to j and is zero otherwise
Q_{ip}	Total quantity of product p while the carried vehicle departing from node i

$$[s_{kj}]_{5 \times 5} = \begin{bmatrix} - & - & - & - & - \\ - & 3 & 1 & 2 & 1 \\ - & 3 & 4 & 3 & 2 \\ - & 4 & 4 & 4 & 3 \\ - & 4 & 4 & 4 & 4 \end{bmatrix}. \quad (9)$$

Since the first location is fixed as the depot, the corresponding probability values of μ_{kj} for the second position of each customer in $N_1 = [1, 2, 3, 4]$ are ($\mu_{21} = (3/(3+1+2+1)) = 0.43$), ($\mu_{22} = (1/(3+1+2+1)) = 0.14$), ($\mu_{23} = (2/(3+1+2+1)) = 0.29$), and ($\mu_{24} = (1/(3+1+2+1)) = 0.14$).

Then, the pheromone matrix is normalized by formula (10). Finally, let (σ_{kj}) denote the probability of customer j at position k which is calculated by formula (11). Based on this value, the ant selects the next customer to visit:

$$\zeta_{ij} = \frac{\tau_{ij}}{\sum_{l \in N_k} \tau_{il}}, \quad (10)$$

$$\sigma_{kj} = \mu_{kj} + \zeta_{ij}, \quad (i = 2, 3, \dots, n). \quad (11)$$

3.2.2. Initialization of the Probabilistic Model Based on a Heuristic Rule. As mentioned in the introduction, normally, the probabilistic model (that is the pheromone matrix in ACO) is initialized to the same value at the beginning of the algorithm. In order to achieve good solutions more quickly, this paper adopts the initial solutions to initialize the proposed probabilistic model but not the same value.

Sort the distance between all customers and the depot from small to large, and take the top $t = \text{round}(n/(10))$ customers into set c^{first} . Let $c^{\text{first}} = [c_1^{\text{first}}, c_2^{\text{first}}, \dots, c_t^{\text{first}}]$ denote the first customer to be visited by the t -th ant, $R = [R_1, R_2, \dots, R_t]$ the complete path passed by the t -th ant (also the t -th initial solution), and $L = [L_1, L_2, \dots, L_t]$ the length of the route taken by the t -th ant. The procedure of the probabilistic model initialization is given as follows:

Step 1: set $t = 1$ and $(\tau_{ij}^0 = \text{zeros}(n \times n))$ and $(s_{kj}^0 = \text{zeros}(n \times n))$.

Step 2: the ant starts from the depot 0 and visits the first customer c_t^{first} . Then, the ant selects the remaining customers for access based on the nearest neighbor principle. This process forms the t -th initial solution R_t .

Step 3: calculate the route length L_t corresponding to R_t .

Step 4: set $s_{kj}^t = s_{kj}^{t-1} + s_{kj}(R_t)$.

If $t = 1$, then set $\tau_{ij}^1 = (1/L_1)$; otherwise, set $\tau_{ij}^t = \tau_{ij}^{t-1} + (1/L_{t1})$.

Step 5: set $t = t + 1$.

Step 6: if $t \leq \text{round}(n/10)$, then go to step 2.

Step 7: output s_{kj}^t and τ_{ij}^t .

3.3. Ant Path Construction. In this study, each ant produces a complete solution for the MCVRP. The ant begins from the depot and continuously chooses customers to its tour as long as the constraint of capacity is satisfied. If there are no feasible customers to select, the ant comes back to the depot and restarts another trip.

Let m denote the amounts of ants and t the amounts of the first customers selected. Set $(t = 0, 1, \dots, m)$. To guarantee the diversity of individuals, the first customer is picked randomly from $n - t$ customers. For example, the first ant randomly selects one of the n customers as the first one to visit. Then, the first customer to be visited by the second ant will be randomly selected from the remaining $n - 1$ customers. If $(m > n)$, when $(t = n + 1)$, reset t to zero.

Afterwards, each ant proceeds to add customers to its route based on the probability function (12). The probability of choosing customer j as the next one in the t -th iteration is given by formula (13):

$$\delta_{ij} = (\sigma_{kj})^\alpha (\eta_{ij})^\beta. \quad (12)$$

In formula (12), σ_{kj} contains information about how often the customer j appears in the current position k , η_{ij} is the inverse of distance between customer i and j , and α and β denote the importance of σ_{kj} and η_{ij} , respectively:

$$p_{ij}^t = \begin{cases} \frac{\delta_{ij}}{\sum_{l \in N_k} \delta_{il}}, & \text{if } j \in N_k, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

Set q_0 be a constant value and q a random variable which is uniformly distributed in $[0, 1]$. If $q > q_0$, select the next customer in accordance with the formulation of p_{ij}^t ; otherwise, select the customer who has the maximum value of δ_{ij} [22].

3.4. Update Mechanism of Probability Matrices. The probabilistic model consists of two probability matrices s_{kj} and τ_{ij} . They are updated by the best solution found at present. Let ρ ($0 \leq \rho < 1$) be the pheromone persistence, then equation (14) is used to update the pheromone concentration in the path. Meanwhile, equation (15) is adopted to update the probability matrix based on the order of customers:

$$\tau_{ij}^{t+1} = \begin{cases} \rho \times \tau_{ij}^t + (1/L_{\text{best}}^t), & \text{if } \text{arc}(i, j) \in L_{\text{best}}^t, \\ \rho \times \tau_{ij}^t, & \text{otherwise.} \end{cases} \quad (14)$$

In equation (14), L_{best}^t is the total length corresponding to the optimal ant path in the t -th iteration:

$$s_{kj}^{t+1} = s_{kj}^t + s_{kj}(\pi_{\text{best}}^t). \quad (15)$$

In equation (15), $s_{kj}(\pi_{\text{best}}^t)$ is the probability matrix s_{kj} obtained by the t -th optimal individual π_{best}^t . We refer to Section 3.2.1 for more details.

3.5. Speed-Up VND-Based Local Search. Generally, for the combinatorial optimization problems, e.g., MCVRP, there are many other high-quality solutions near the high-quality solutions found so far. The algorithm in this section performs the problem-dependent local search on the global optimal solution $\pi_{\text{best}}^{\text{gen}}$, for the purpose of realizing the detailed search of the area near $\pi_{\text{best}}^{\text{gen}}$. In this paper, the operations of insert, swap, and intersection elimination based on VND, the speed-up search strategy, and the first move strategy are adopted.

3.5.1. Basic and Pipe VND. The VND is a variant of the variable neighborhood search. The basic VND [33] explores the list of neighborhoods in a sequential way one by one. Once an improvement is detected in any neighborhood structure, the search goes back to the first neighborhood from the list. Until all the neighborhoods encounter the local optimum, the search ends. However, the pipe VND [33] executes continuous search in the same neighborhood if an improvement is obtained. In this paper, both these VNDs are applied.

3.5.2. Operation of Insert and Swap. Assume an ant's tour is represented as $\pi_{\text{ant}}^{\text{old}} = [1, \dots, i-1, i, i+1, \dots, j-1, j, j+1, \dots, n]$. The insert operator selects one customer like i and inserts it before or after another customer like j . Then, the tour becomes $\pi_{\text{ant}}^{\text{new_insert}} = [1, \dots, i-1, i+1, \dots, j-1, i, j, j+1, \dots, n]$ or $\pi_{\text{ant}}^{\text{new_insert}} = [1, \dots, i-1, i+1, \dots, j-1, j, i, j+1, \dots, n]$. The swap operator selects two customers like i and j on this tour and swaps them. Then, $\pi_{\text{ant}}^{\text{new_swap}} = [1, \dots, i-1, j, i+1, \dots, j-1, i, j+1, \dots, n]$.

3.5.3. Operation of Intersection Elimination. Since the objective function of the MCVRP is constructed to minimize the total length of the path, the optimal path of each vehicle should not be crossed.

Theorem 1. *In the route of every vehicle, if two lines intersect, the total path length of the vehicle is larger than that of the total length without intersection.*

Proof. As shown in Figure 1, there is one vehicle route denoted by $\pi_{\text{ant_vehicle}}^{\text{old}} = [0, 1, 3, 2, 4, 0]$. For any two given intersecting line segments, such as lines L_{13} and L_{24} which intersect at point u , the end points of the two segments that can form the opposite side are connected like L_{12} and L_{34} . Next, the new path without intersection is obtained and denoted by $\pi_{\text{ant_vehicle}}^{\text{new}} = [0, 1, 2, 3, 4, 0]$. According to the theorem that the sum of any two sides in a triangle is greater than the third side, in $(\Delta 1u2)$, $(L_{1u} + L_{2u} > L_{12})$; in $\Delta 3u4$, $(L_{u3} + L_{u4} > L_{34})$. Let $L_{\text{total}}^{\text{old}}$ be the total length of vehicle path with intersecting lines, then $L_{\text{total}}^{\text{old}} = L_{01} + L_{1u} + L_{u3} + L_{32} + L_{2u} + L_{u4} + L_{40}$. Let $L_{\text{total}}^{\text{new}}$ be the total length after eliminating intersecting lines, then $L_{\text{total}}^{\text{new}} = L_{01} + L_{12} + L_{23} + L_{34} + L_{40}$. As a result, $L_{\text{total}}^{\text{new}} - L_{\text{total}}^{\text{old}} = [L_{12} - (L_{1u} + L_{2u})] + [L_{34} - (L_{u3} + L_{u4})] < 0$.

From $\pi_{\text{ant_vehicle}}^{\text{old}}$ and $\pi_{\text{ant_vehicle}}^{\text{new}}$ defined above, it can be seen that if two paths intersect, there are four customers involved. According to the order of customers visited by the vehicle, only exchange positions of the second and third customers and then cross paths can be eliminated. In this paper, the geometric method is used to determine whether the crossing occurs.

Suppose the coordinates of customers 1, 3, 2, and 4 in Figure 1 are (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , and (x_4, y_4) , and then, the linear equations of line segments L_{13} and L_{24} are shown in the following equations:

$$\begin{cases} x = x_1 + \lambda(x_2 - x_1), \\ y = y_1 + \lambda(y_2 - y_1), \end{cases} \quad (16)$$

$$\begin{cases} x = x_3 + \psi(x_4 - x_3), \\ y = y_3 + \psi(y_4 - y_3), \end{cases} \quad (17)$$

$$\begin{cases} (x_2 - x_1)\lambda + (x_3 - x_4)\psi = x_3 - x_1, \\ (y_2 - y_1)\lambda + (y_3 - y_4)\psi = y_3 - y_1. \end{cases} \quad (18)$$

In formulas (16) and (17), λ is the slope of the line where L_{13} is located, and ψ is the slope of the line where L_{24} is located.

If L_{13} and L_{24} intersect, then formula (18) has a solution. In other words, inequality (19) should be satisfied. Meanwhile, it is necessary to satisfy the inequality (22) in order to make two line segments intersect within the line segment (this is the case of path intersection shown in Figure 1), rather than the vertex of the line segment or its extension line. However, equations (20) and (21) are used to calculate λ and ψ , respectively:

$$\Delta = \begin{vmatrix} x_2 - x_1 & x_3 - x_4 \\ y_2 - y_1 & y_3 - y_4 \end{vmatrix} \neq 0, \quad (19)$$

$$\lambda = \frac{1}{\Delta} \times \begin{vmatrix} x_3 - x_1 & x_3 - x_4 \\ y_3 - y_1 & y_3 - y_4 \end{vmatrix}, \quad (20)$$

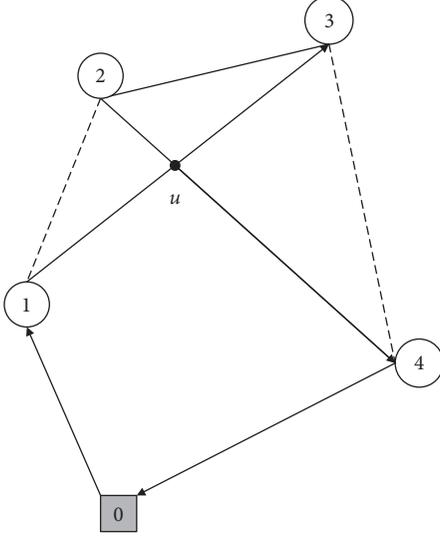


FIGURE 1: Route of one vehicle.

$$\psi = \frac{1}{\Delta} \times \begin{vmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{vmatrix}, \quad (21)$$

$$\begin{cases} 0 < \lambda < 1, \\ 0 < \psi < 1. \end{cases} \quad (22)$$

□

$$N_{\text{insert}}(\pi, i) = \{\pi^{i,j} = \text{insert}(\pi, i, j) \mid i = 1, 2, \dots, n-1, j = i+1, \dots, n, \text{ and } (i, j) \notin I_{\text{insert}}^{\text{infeasible}}\}, \quad (23)$$

$$N_{\text{swap}}(\pi, i) = \{\pi^{i,j} = \text{swap}(\pi, i, j) \mid i = 1, 2, \dots, n-1, j = i+1, \dots, n, \text{ and } (i, j) \notin I_{\text{swap}}^{\text{infeasible}}\}, \quad (24)$$

$$N_{\text{elim_intersection}}(\pi, k) = \{\pi^{i,j} = \text{elim_intersection}(\pi, i, j) \mid i = l, j = i+1, \dots, i+n_k-1\}. \quad (25)$$

It can be seen from (23)–(25) that $N_{\text{insert}}(\pi, i)$ is a feasible neighborhood set that any customer i inserts before or after any possible customer j within the same vehicle or between different vehicles. $N_{\text{swap}}(\pi, i)$ is a feasible neighborhood set that any customer i swaps with any possible customer j within the same vehicle or between different vehicles. $N_{\text{elim_intersection}}(\pi, k)$ is a feasible neighborhood set that customers involved in the cross-line can be interchanged within the k -th vehicle.

3.5.4. Speed-Up Search Strategy. Let $\text{insert}(\pi, i, j)$, $\text{swap}(\pi, i, j)$, and $\text{elim_intersection}(\pi, i, j)$ represent the operations of inserting, swapping, and uncrossing the i -th and j -th elements in solution π , respectively. Let $I_{\text{insert}}^{\text{infeasible}}$ denote the set of (i, j) whose neighborhood is infeasible (the infeasibility here means that the vehicle capacity constraint is not satisfied) after executing $\text{insert}(\pi, i, j)$, and $I_{\text{swap}}^{\text{infeasible}}$ is the set of (i, j) whose neighborhood is infeasible after executing $\text{swap}(\pi, i, j)$. Because the intersection elimination operation is only carried out within the customers served by each vehicle, there is no neighborhood that violates the capacity constraint. Let $N_{\text{insert}}(\pi, i)$ denote the feasible neighborhood set based on insert operation for i customers in solution π , $N_{\text{swap}}(\pi, i)$ the feasible neighborhood set based on swap operation for i customers in solution π , and $N_{\text{elim_intersection}}(\pi, k)$ the feasible neighborhood set based on intersection elimination operation for customers serviced by the k -th vehicle in solution π . Set n_k to be the number of customers visited by the k -th vehicle, then $(l = \sum_{i=1}^k n_i - n_k + k + 1)$ is the position of the first customer visited by the k -th vehicle in solution π . Hence, there are as follows:

For the MCVRP, if customer j is too far away from customer i , the total route distance after the insert and swap operation will be greater than before, which makes the two operations meaningless. Hence, in order to avoid the search of invalid areas, the distances between all possible customers j and customer i are arranged from small to large, and just the first S points are taken out to insert and swap. Thus, the more compact neighborhood sets $N_{\text{insert}}^{\text{narrow}^1}(\pi, i)$ and $N_{\text{swap}}^{\text{narrow}^1}(\pi, i)$ can be obtained:

$$\begin{aligned} N_{\text{insert}}^{\text{narrow}^1}(\pi, i) &= \{\pi^{i,j} = \text{insert}(\pi, i, j) \mid i = 1, 2, \dots, n-1, j \in S, \text{ and } (i, j) \notin I_{\text{insert}}^{\text{infeasible}}\}, \\ N_{\text{swap}}^{\text{narrow}^1}(\pi, i) &= \{\pi^{i,j} = \text{swap}(\pi, i, j) \mid i = 1, 2, \dots, n-1, j \in S, \text{ and } (i, j) \notin I_{\text{swap}}^{\text{infeasible}}\}. \end{aligned} \quad (26)$$

For the sake of speeding up the searching process and making the neighborhood search focus on high-quality areas, the nature of the problem can be utilized to further decrease invalid operations and reduce the feasible neighborhood set. Let

$L(\pi)$ be the value of the objective function which is the total mileage of the route corresponding to solution π and $d_{\pi[i], \pi[j]}$ be the distance between element $\pi[i]$ which is located in position i of π and element $\pi[j]$ which is located in position j of π .

For $N_{\text{insert}}^{\text{narrow}1}(\pi, i)$, if customer i is inserted before customer j , $L(\pi^{i,j}) - L(\pi) = d(\pi^{i,j}) - d(\pi) = (d_{\pi[i-1],\pi[i+1]} + d_{\pi[j-1],\pi[j]} + d_{\pi[i],\pi[j]}) - (d_{\pi[i-1],\pi[i]} + d_{\pi[i],\pi[i+1]} + d_{\pi[j-1],\pi[j]})$; if customer i is inserted after customer j , $L(\pi^{i,j}) - L(\pi) = d(\pi^{i,j}) - d(\pi) = (d_{\pi[i-1],\pi[i+1]} + d_{\pi[j],\pi[i]} + d_{\pi[i],\pi[j+1]}) - (d_{\pi[i-1],\pi[i]} + d_{\pi[i],\pi[i+1]} + d_{\pi[j],\pi[j+1]})$. For any (i, j) , if $d(\pi^{i,j}) - d(\pi) > 0$, then $L(\pi^{i,j}) > L(\pi)$. (i, j) in this case can

$$N_{\text{insert}}^{\text{narrow}2}(\pi, i) = \{\pi^{i,j} = \text{insert}(\pi, i, j) | i = 1, 2, \dots, n-1, j \in S, (i, j) \notin IJ_{\text{insert}}^{\text{infeasible}}, \text{ and } d(\pi^{i,j}) - d(\pi) < 0\}. \quad (27)$$

For $N_{\text{swap}}^{\text{narrow}1}(\pi, i)$, $L(\pi^{i,j}) - L(\pi) = d(\pi^{i,j}) - d(\pi) = (d_{\pi[i-1],\pi[j]} + d_{\pi[j],\pi[i+1]} + d_{\pi[j-1],\pi[i]} + d_{\pi[i],\pi[j+1]}) - (d_{\pi[i-1],\pi[i]} + d_{\pi[i],\pi[i+1]} + d_{\pi[j-1],\pi[j]} + d_{\pi[j],\pi[j+1]})$. And for any (i, j) , if $d(\pi^{i,j}) - d(\pi) > 0$, then $L(\pi^{i,j}) > L(\pi)$. (i, j) in this case can be directly excluded without calculating $L(\pi^{i,j})$. If

$$N_{\text{swap}}^{\text{narrow}2}(\pi, i) = \{\pi^{i,j} = \text{swap}(\pi, i, j) | i = 1, 2, \dots, n-1, j \in S, (i, j) \notin IJ_{\text{swap}}^{\text{infeasible}}, \text{ and } d(\pi^{i,j}) - d(\pi) < 0\}. \quad (28)$$

For $N_{\text{elim_intersection}}(\pi, k)$, we can use the method described in Section 3.5.3 to determine whether two paths intersect. If so, according to Theorem 1, we know $L(\pi^{i,j}) - L(\pi) = d(\pi^{i,j}) - d(\pi) = (d_{\pi[i-1],\pi[j]} + d_{\pi[i],\pi[j+1]}) - (d_{\pi[i-1],\pi[i]} + d_{\pi[i],\pi[j+1]}) < 0$. Then, $L(\pi^{i,j})$ can be obtained directly from $L(\pi^{i,j}) = L(\pi) + (d(\pi^{i,j}) - d(\pi))$ instead of starting from the first customer of π to calculate.

Let $\text{SP_FindBest}N_{\text{insert}}^{\text{narrow}2}(\pi, i)$, $\text{SP_FindBest}N_{\text{swap}}^{\text{narrow}2}(\pi, i)$, and $\text{SP_FindBest}N_{\text{elim_intersection}}(\pi, k)$ be the optimal neighborhoods in searching and outputting $N_{\text{insert}}^{\text{narrow}2}(\pi, i)$, $N_{\text{swap}}^{\text{narrow}2}(\pi, i)$, and $N_{\text{elim_intersection}}(\pi, k)$, respectively. These three optimal neighborhoods utilize the above speed-up search method in the local search of IHACO. Meanwhile, the pipe VND described in Section 3.5.1 is also applied. The corresponding procedures are given as follows:

- Step 1: set $i = 1$ and $\pi_{\text{best}}^{\text{previous}} = \pi_{\text{best}}^{\text{gen}}$.
- Step 2: $\pi_{\text{local_best}}^i = \text{SP_FindBest}N_{\text{insert}}^{\text{narrow}2}(\pi_{\text{best}}^{\text{gen}}, i)$.
- Step 3: if $L(\pi_{\text{local_best}}^i) < L(\pi_{\text{best}}^{\text{gen}})$, then $\pi_{\text{best}}^{\text{gen}} = \pi_{\text{local_best}}^i$.
- Step 4: if $i < n-1$, then $i = i+1$ and go to step 2.
- Step 5: if $i = n-1$, then $\pi_{\text{best}}^{\text{current}} = \pi_{\text{best}}^{\text{gen}}$.
- If $\pi_{\text{best}}^{\text{current}} < \pi_{\text{best}}^{\text{previous}}$, go to step 1.
- Step 6: set $i = 1$, and $\pi_{\text{best}}^{\text{previous}} = \pi_{\text{best}}^{\text{gen}}$.
- Step 7: $\pi_{\text{local_best}}^i = \text{SP_FindBest}N_{\text{swap}}^{\text{narrow}2}(\pi_{\text{best}}^{\text{gen}}, i)$.
- Step 8: if $L(\pi_{\text{local_best}}^i) < L(\pi_{\text{best}}^{\text{gen}})$, then $\pi_{\text{best}}^{\text{gen}} = \pi_{\text{local_best}}^i$.
- Step 9: if $i < n-1$, then $i = i+1$ and go to step 7.
- Step 10: if $i = n-1$, then $\pi_{\text{best}}^{\text{current}} = \pi_{\text{best}}^{\text{gen}}$.
- If $\pi_{\text{best}}^{\text{current}} < \pi_{\text{best}}^{\text{previous}}$, go to step 6.
- Step 11: set $k = 1$ and $\pi_{\text{best}}^{\text{previous}} = \pi_{\text{best}}^{\text{gen}}$.
- Step 12: $\pi_{\text{local_best}}^k = \text{SP_FindBest}N_{\text{elim_intersection}}(\pi_{\text{best}}^{\text{gen}}, k)$.
- Step 13: if $L(\pi_{\text{local_best}}^k) < L(\pi_{\text{best}}^{\text{gen}})$, then $\pi_{\text{best}}^{\text{gen}} = \pi_{\text{local_best}}^k$.

be directly excluded without calculating $L(\pi^{i,j})$. If $L(\pi^{i,j}) < L(\pi)$, then $L(\pi^{i,j})$ can be obtained directly from $L(\pi^{i,j}) = L(\pi) + (d(\pi^{i,j}) - d(\pi))$ instead of starting from the first customer of π to calculate. Therefore, a more compact and effective neighborhood set $N_{\text{insert}}^{\text{narrow}2}(\pi, i)$ can be obtained on the basis of $N_{\text{insert}}^{\text{narrow}1}(\pi, i)$:

$L(\pi^{i,j}) < L(\pi)$, then $L(\pi^{i,j})$ can be obtained directly from $L(\pi^{i,j}) = L(\pi) + (d(\pi^{i,j}) - d(\pi))$ instead of starting from the first customer of π to calculate. Therefore, a more compact and effective neighborhood set $N_{\text{swap}}^{\text{narrow}2}(\pi, i)$ can be obtained on the basis of $N_{\text{swap}}^{\text{narrow}1}(\pi, i)$:

Step 14: if $k < K$, then $k = k+1$ and go to step 12.

Step 15: if $k = K$, then $\pi_{\text{best}}^{\text{current}} = \pi_{\text{best}}^{\text{gen}}$.

If $\pi_{\text{best}}^{\text{current}} < \pi_{\text{best}}^{\text{previous}}$, go to step 11.

Step 16: output $\pi_{\text{best}}^{\text{gen}}$.

3.5.5. First Move Strategy. In order to reach more different regions, a first move strategy is adopted in $\text{SP_FindBest}N_{\text{insert}}^{\text{narrow}2}(\pi, i)$, $\text{SP_FindBest}N_{\text{swap}}^{\text{narrow}2}(\pi, i)$, and $\text{SP_FindBest}N_{\text{elim_intersection}}(\pi, k)$. Here is an example to illustrate the strategy. While the first neighbor π_{first}^i in $N_{\text{insert}}^{\text{narrow}2}(\pi, i)$ which can improve $\pi_{\text{best}}^{\text{gen}}$ is gained, $\text{SP_FindBest}N_{\text{insert}}^{\text{narrow}2}(\pi, i)$ terminates and outputs π_{first}^i . Furthermore, the sequence π in $\text{SP_FindBest}N_{\text{insert}}^{\text{narrow}2}(\pi, i)$ is replaced by $\pi_{\text{best}}^{\text{gen}}$ for the purpose of implementing the search instantly from the promising areas discovered by $\text{SP_FindBest}N_{\text{insert}}^{\text{narrow}2}(\pi, i)$. Obviously, the algorithm will search in the vicinity of varied suboptimal solutions, but not limited to the vicinity of the optimal solution, thus expanding the scope of the search area. The changed procedures with the above strategy, named $\text{SP_FindFirstMove}N_{\text{insert}}^{\text{narrow}2}(\pi, i)$, $\text{SP_FindFirstMove}N_{\text{swap}}^{\text{narrow}2}(\pi, i)$, and $\text{SP_FindFirstMove}N_{\text{elim_intersection}}(\pi, k)$, are only different from $\text{SP_FindBest}N_{\text{insert}}^{\text{narrow}2}(\pi, i)$, $\text{SP_FindBest}N_{\text{swap}}^{\text{narrow}2}(\pi, i)$, and $\text{SP_FindBest}N_{\text{elim_intersection}}(\pi, k)$ in step 2, step 7, and step 12. The pseudocodes of the changed steps are as follows:

- Step 2: $\pi_{\text{local_best}}^i = \text{SP_FindFirstMove}N_{\text{insert}}^{\text{narrow}2}(\pi_{\text{best}}^{\text{gen}}, i)$.
- Step 7: $\pi_{\text{local_best}}^i = \text{SP_FindFirstMove}N_{\text{swap}}^{\text{narrow}2}(\pi_{\text{best}}^{\text{gen}}, i)$.
- Step 12: $\pi_{\text{local_best}}^k = \text{SP_FindFirstMove}N_{\text{elim_intersection}}(\pi_{\text{best}}^{\text{gen}}, k)$.

3.5.6. Procedure of Local Search. Let $\text{SPVND_FindFirstMove}N_{\text{insert}}^{\text{narrow}2}(\pi)$, $\text{SPVND_FindFirstMove}N_{\text{swap}}^{\text{narrow}2}(\pi)$, and $\text{SPVND_FindFirstMove}N_{\text{elim_intersection}}(\pi)$ denote

SP_FindFirstMove $N_{\text{insert}}^{\text{narrow}2}(\pi)$, SP_FindFirstMove $N_{\text{swap}}^{\text{narrow}2}(\pi)$, and SP_FindFirstMove $N_{\text{elim_intersection}}(\pi)$ with the pipe VND method. Moreover, the basic VND described in Section 3.5.1 is also adopted. Then, the procedure of the IHACO's local search is given in the following:

Step 1: Perturbation phase.

Set $\pi^0 = \pi_{\text{best}}^{\text{gen}}$;

For $kk = 1$ to KP

Randomly select i and $j (i \neq j)$;

$\pi^1 = \text{insert}(\pi^0, i, j)$;

$\pi^0 = \pi^1$;

End.

Step 2: exploitation phase.

Set $\text{loop} = 0$;

Repeat

$\pi^2 = \text{SPVND_FindFirstMove}N_{\text{insert}}^{\text{narrow}2}(\pi^1)$;

$\pi^3 = \text{SPVND_FindFirstMove}N_{\text{swap}}^{\text{narrow}2}(\pi^2)$;

If $\pi^3 < \pi^2$ then $\pi^1 = \pi^3$ and

go to $\pi^2 = \text{SPVND_FindFirstMove}N_{\text{insert}}^{\text{narrow}2}(\pi^1)$;

else

$\pi^4 = \text{SPVND_FindFirstMove}$

$N_{\text{elim_intersection}}(\pi^3)$;

If $\pi^4 < \pi^3$ then $\pi^1 = \pi^4$ and

go to $\pi^2 = \text{SPVND_FindFirstMove}N_{\text{insert}}^{\text{narrow}2}(\pi^1)$;

else

loop ++;

end;

end;

until $\text{loop} = 1$.

Step 3: if $L(\pi^4) < L(\pi_{\text{best}}^{\text{gen}})$, then $\pi_{\text{best}}^{\text{gen}} = \pi^4$.

Step 4: output $\pi_{\text{best}}^{\text{gen}}$.

In the above procedure, step 1 is a perturbation phase, which is helpful to prevent the cycling search and overcome the local optima. The special devised step 2 performs exploitation from the region obtained by step 1.

3.6. *IHACO*. Denote gen as a generation and maxgen as the maximum generation. Based on the above subsections, the proposed IHACO is given in the following:

Step 1: parameter initialization. Set the crucial parameter m , maxgen , α , β , ρ , ρ_0 , S , and KP .

Step 2: probabilistic model initialization. Set $\text{gen} = 1$ //Section 3.2.2

Step 3: ant path construction. Generate the population at generation gen by sampling σ_{kj} and then evaluate its individuals to obtain the global best individual $\pi_{\text{best}}^{\text{gen}}$ //Section 3.3

Step 4: local exploitation. Execute three types of local search to $\pi_{\text{best}}^{\text{gen}}$ and update $\pi_{\text{best}}^{\text{gen}}$ //Section 3.5

Step 5: update mechanism of probability matrices. Calculate the objective function value $L(\pi_{\text{best}}^{\text{gen}})$ corresponding to the best individual $\pi_{\text{best}}^{\text{gen}}$. Update π_{ij} and s_{kj} based on $L(\pi_{\text{best}}^{\text{gen}})$ and $\pi_{\text{best}}^{\text{gen}}$ //Section 3.4

Step 6: set $\text{gen} = \text{gen} + 1$. If $\text{gen} \leq \text{maxgen}$, go to step 3.

Step 7: output the best solution and its objective value found so far.

As it can be seen from the above steps, not only does IHACO takes advantage of the new probabilistic model-based global search to execute exploration to obtain promising regions within solution space but also it makes use of the speed-up VND-based local search with the first move strategy to perform exploitation from these promising regions. On account of both global and local search are well balanced, the IHACO is expected to acquire good results.

4. Computational Experiments and Statistical Analysis

In this section, three sets of numerical simulations are conducted with the aim of checking the effectiveness and efficiency of IHACO. The experiments implemented have been coded in Matlab 2019a and tested on an Intel 1.8 GHz PC with 12 GB RAM.

4.1. *Experimental Setup*. The generation method of instances is derived from [13]. The test set in this paper contains 14 benchmark instances with the capacity constraint, including vrpcn1-vrpcn5 and vrpcn11-vrpcn12 of sets a and b . These scales of problem range from 50 to 200 customers.

The main parameters of IHACO are the same as those in [22]: the number of ants $m = 20$, two exponential factors $\alpha = 1$ and $\beta = 2$, the pheromone concentration $\rho = 0.9$, and the constant value $q_0 = 0.9$. There are other parameters needed to set: the upper bound of range in local search $S = (n/5)$ and the perturbation variable $KP = (n/10)$. For the sake of having a fair comparison, all the compared algorithms terminate when either of the following two criteria is reached: the first one is the same runtime limit, and the second one is less than or equal to the optimal value found in the literature. It should be noted that when setting the termination time of the proposed algorithm, two steps are adopted. We first find the running time (t_{best}) and the running environment (that is, the CPU frequency (f_{best})) corresponding to the best solution of the four comprised algorithms in each instance. Suppose that the CPU frequency of this paper is f , then the termination time of

TABLE 2: Comparison results of IHACO and four algorithms.

Inst.	n	ACS		HAC		HAVNS		HABC		IHACO	
		Obj.	$t(s)$								
vrpnc1a	50	<i>569.564</i>	16	550.70	5	550.7	4	550.42	5	550.70	6
vrpnc1b	50	569.118	17	551.94	5	506.18	4	502.83	6	548.77	7
vrpnc2a	75	957.525	36	890.68	15	881.84	14	890.68	17	873.57	13
vrpnc2b	75	954.856	35	<i>918.96</i>	14	<i>918.96</i>	14	914	17	888.38	15
vrpnc3a	100	964.132	122	874.07	40	878.83	21	<i>875.24</i>	27	870.11	35
vrpnc3b	100	959.327	122	895.26	44	898.64	20	895.39	25	871.22	16
vrpnc4a	150	1253.86	345	1126.12	146	1126.12	103	1126.12	110	1110.06	33
vrpnc4b	150	<i>1254.51</i>	336	1159.48	151	1159.48	120	1159.48	142	1134.90	116
vrpnc5a	199	1587.02	688	1444.29	257	1408.86	230	1385.71	345	<i>1423.11</i>	414
vrpnc5b	199	1640.59	676	1525.87	236	<i>1515.25</i>	218	1492.02	212	1456.16	125
vrpnc11a	120	1133.88	281	1110.45	75	1110.45	70	1110.45	82	1110.45	75
vrpnc11b	120	1247.49	280	<i>1221.73</i>	87	<i>1221.73</i>	77	1220.43	95	1218.66	71
vrpnc12a	100	911.861	105	912.64	15	901.22	13	901.15	35	<i>907.73</i>	42
vrpnc12b	100	970.833	100	950.79	30	935.62	22	923.25	43	961.74	51
Average		1069.612	225.64	1009.5	80	<i>1000.99</i>	66.43	996.23	82.93	994.68	72.79

TABLE 3: Comparison results of IHACO and its variants.

Inst.	n	IHACO_V1	IHACO_V2	IHACO_V3	IHACO_V4	IHACO_V5	IHACO_V6	IHACO
vrpnc1a	50	550.70	550.70	553.44	552.19	550.70	550.70	550.70
vrpnc1b	50	553.48	551.94	557.08	548.77	548.77	548.77	548.77
vrpnc2a	75	888.63	896.90	887.90	896.01	888.69	895.42	873.57
vrpnc2b	75	891.71	905.46	920.60	916.60	<i>900.64</i>	908.29	888.38
vrpnc3a	100	872.29	876.28	885.64	894.72	882.26	<i>875.06</i>	870.11
vrpnc3b	100	<i>879.28</i>	874.96	893.21	897.09	881.33	882.38	871.22
vrpnc4a	150	<i>1115.58</i>	1111.04	1143.92	1157.33	1125.19	1127.24	1110.06
vrpnc4b	150	1151.64	1144.28	1167.45	1174.29	1151.07	<i>1148.68</i>	1134.90
vrpnc5a	199	1444.35	<i>1428.19</i>	1494.80	1486.06	1424.34	1460.31	1423.11
vrpnc5b	199	1462.21	1494.86	1531.33	1526.43	<i>1472.54</i>	1505.95	1456.16
vrpnc11a	120	<i>1128.12</i>	1128.94	1172.76	1130.86	1123.60	1133.39	1110.45
vrpnc11b	120	1226.90	1238.30	1243.82	1237.96	<i>1231.04</i>	1239.54	1218.66
vrpnc12a	100	920.09	922.78	932.85	939.67	919.65	<i>919.81</i>	907.73
vrpnc12b	100	975.66	<i>974.97</i>	978.48	993.87	967.61	979.08	961.74
Average		1004.33	1007.12	1025.95	1025.13	<i>1004.82</i>	1012.47	994.68

the proposed algorithm $t = t_{\text{best}} \times (f_{\text{best}}/f)$. In addition, to make the results clearer, in Tables 2 and 3, the best, the second-best, and the third-best values in each row are marked out with the bold, the bold and underlined, and the italic and underlined fonts.

4.2. Comparison of IHACO and Other Algorithms. To test the performance of the algorithm in this paper, various simulations are carried out to compare the proposed IHACO with a classical ACO (ACS) [21], an effective hybridized ACO (HAC) [22], and two new state-of-the-art approaches named HAVNS and HABC [24] for the MCVRP. In solving MCVRP, ACS is the earliest used ant colony optimization algorithm and is proved with high performance for two-compartment test problems. HAC that is presented through the inspiration of ACS combines the ant colony optimization algorithm with local search schemes. After testing new generated benchmark problem instances, the results manifest that HAC takes less computational time to get higher quality solutions for all given problems except one problem

(vrpnc12a) compared to ACS. HAVNS and HABC are also heuristic-based. According to the experiment results in [24], for the 14 examples to be tested in this paper, HAVNS obtains 12 best solutions and updates 6 of them on the basis of HAC. Meanwhile, HABC gains 12 best solutions and updates 8 of them based on HAC.

The experimental results of the five algorithms are shown in Table 2. The first column in Table 2 represents the type of the instance, and the second column is the quantity of customers contained in every instance. The meaning of *Obj.* and *T(s)* in this table is the objective function value and runtime, respectively.

As reported from Table 2, the proposed IHACO finds the best solution for 9 instances of the 14 problem instances and updates 8 of them based on HAVNS and HABC. However, on the average value of all examples, IHACO performs best. Moreover, while the quantity of customers is more than 100, IHACO can hit 5 best solutions in 6 instances. Hence, as the problem size increases, IHACO is more effective. In addition, HAVNS and HABC were implemented on Intel Core™ i5-5200U @ 2.20 GHz 6,00 GO RAM, which was different

from the IHACO operating environment. Considering the speed of CPU, IHACO is much faster than HAVNS and HABC through the conversion of $((72.79/(2.2/1.8)) = 59.56)$ on average. Furthermore, as it can be seen from Table 2, the time consumption corresponding to the 9 good solutions obtained in this study is less than that of the best solution in three comparative literatures. Considering the difference of computing environment, if the conversion is carried out, the time consumption of these 9 solutions will be shorter. So, IHACO is also more efficient.

4.3. Comparison of IHACO with Its Six Variants. For the purpose of evaluating the crucial operations of IHACO, IHACO is also compared with its six variants, whose abbreviations are described in the following:

- (1) IHACO_V1: in IHACO's global search, s_{kj} is not adopted, and only τ_{ij} is used to realize the generation of new individuals.
- (2) IHACO_V2: in IHACO's global search, when the probabilistic model initialization phase is carried out, τ_{ij} is initialized to the same value $(1/L^{\text{neig}})$, and s_{kj} is initialized just by the individual π^{neig} (the individual corresponding to the objective function value L^{neig}).
- (3) IHACO_V3: in IHACO's local search, $N_{\text{insert}}^{\text{narrow}1}(\pi, i)$ and $N_{\text{swap}}^{\text{narrow}1}(\pi, i)$ are not employed. In other words, all possible values for customer j ($(j \in S)$) is replaced with $(j = i + 1, \dots, n)$ will be assessed.
- (4) IHACO_V4: in IHACO's local search, the first move strategy is not employed. This means that, in the operations of insert and swap, after all possible positions have been tried, and the customer chooses the location that has the highest improvement to insert or swap.
- (5) IHACO_V5: in IHACO's local search, the pipe VND is not employed.
- (6) IHACO_V6: in IHACO's local search, the basic VND is not employed.

The statistical results are listed in Table 3. Besides, the graphical results for the compared algorithms are exhibited in Figure 2.

Compared with IHACO's variants, it is clear from Table 3 and Figure 2 that IHACO performs best for all instances. The results indicate the importance of adopting all the crucial operations of IHACO simultaneously. It can also be found from Figure 2 that, as the problem scale increases, the difference of the objective values obtained by these comparison algorithms gradually increases, and then, the superiority of IHACO becomes more obvious. This once again verified a conclusion of the previous experiment; that is, IHACO performs better when the number of customers is large.

The above results can be further analysed. IHACO_V1 does not adopt the information sequence matrix, which may lead to the low accuracy of the information left by ants on the

path and reduces the similarity between the sampled individuals and the optimal individuals. IHACO_V2 initializes the two matrixes that make up the probabilistic model to a unified value. As a result, the initial individuals have no information of excellent individuals to utilize, and then, the quality of the initial population is lowered. IHACO_V3 expands the range of invalid regions in operations of insert and swap. As a result, it could take a longer time to find high-quality solutions than IHACO. IHACO_V4 must find the optimal location to execute the insert and swap operations. This causes that the search areas are relatively concentrated, which may miss some promising regions. Both IHACO_V5 and IHACO_V6 reduce the depth of exploitation, so it is not easy for them to achieve high-quality solutions. Thus, the main operations removed from these six variants are all adopted by IHACO.

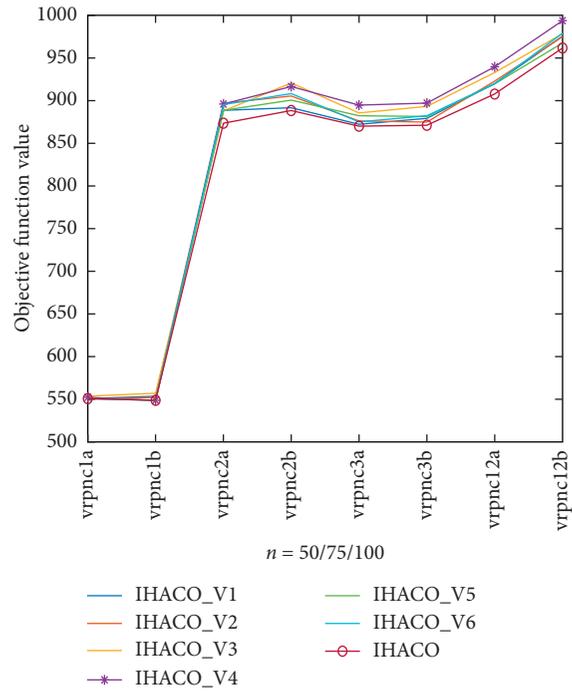
4.4. Superiority Verification of Using Multi-Compartment Vehicles. Since some different products have different characteristic properties, they cannot be mixed. That is to say, the products cannot be transported simultaneously by vehicles with single compartment. However, they can be transported by vehicles with single compartment for several times or by vehicles with multiple compartments at one time.

In this section, for the two kinds of products involved in the problem instances adopted in this paper, single-compartment vehicles and two-compartment vehicles are both utilized to gather the needs of customers within the service area. When using single-compartment vehicles to collect goods from customers, the problem is decomposed into two subproblems. One is to seek the solution of VRP for collecting product 1, and another is for collecting product 2. The sum of the total distance for the two subproblems represents the final solution listed in column 3 of Table 4. The results obtained by using two-compartment vehicles are the same as reported in column 5 of Table 2. The increase percentage of total length listed in column 5 of Table 4 can be obtained from the following equation:

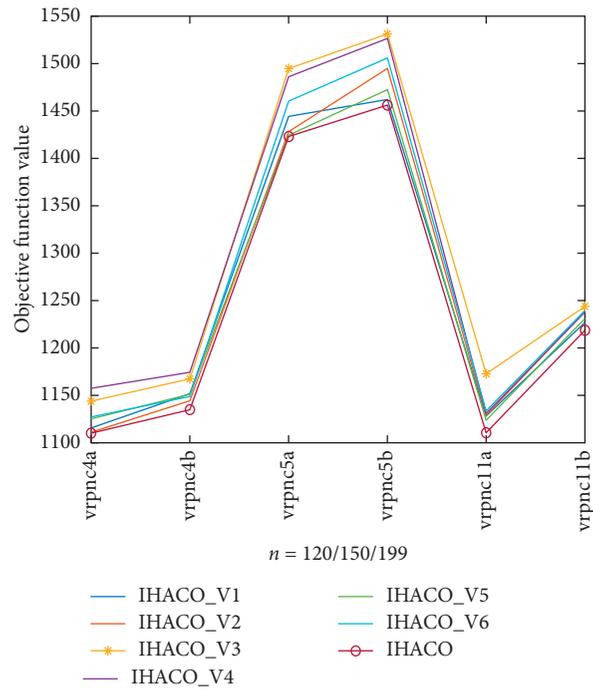
$$\text{increase\%} = \frac{\text{SC} - \text{TC}}{\text{SC}} \times 100, \quad (29)$$

where SC and TC are the total distance travelled by single-compartment vehicles and two-compartment vehicles, respectively. Besides, in order to make the comparison more intuitive, plot with 95% Tukey's honest significant difference (HSD) confidence intervals for employing these two types of compartments under two categories of data sets is shown in Figure 3.

It can be obviously seen from Table 4 and Figure 3 that using single-compartment vehicles brings the notable increase in the total length of each route. The reason is that although the total route cost is reduced due to the increase of vehicle capacity while transporting the same product, each customer must be served twice in order to ensure that different products are all delivered. More specifically, it can be noticed from Table 4 that the total distance raises 47.72% on average when the vehicles with the single compartment are employed to visit the customers. Moreover, it is worth



(a)



(b)

FIGURE 2: Comparison results of IHACO and its variants on different-scaled instances: (a) the customer scale is no more than 100; (b) the customer scale is greater than 100.

TABLE 4: Comparison results of using different compartment vehicles.

Inst.	n	Total length of using single compartment	Total length of using two compartments	Total length increase percentage (%)
vrpnc1a	50	932.13	550.70	69.26
vrpnc1b	50	934.13	548.77	70.22
vrpnc2a	75	1317.96	873.57	50.87
vrpnc2b	75	1327.43	888.38	49.42
vrpnc3a	100	1423.90	870.11	63.65
vrpnc3b	100	1415.06	871.22	62.42
vrpnc4a	150	1709.25	1110.06	53.98
vrpnc4b	150	1690.64	1134.90	48.97
vrpnc5a	199	2043.76	1423.11	43.61
vrpnc5b	199	2049.47	1456.16	40.74
vrpnc11a	120	1583.43	1110.45	42.59
vrpnc11b	120	1556.18	1218.66	27.70
vrpnc12a	100	1281.69	907.73	41.20
vrpnc12b	100	1305.50	961.74	35.74
Average		1469.32	994.68	47.72

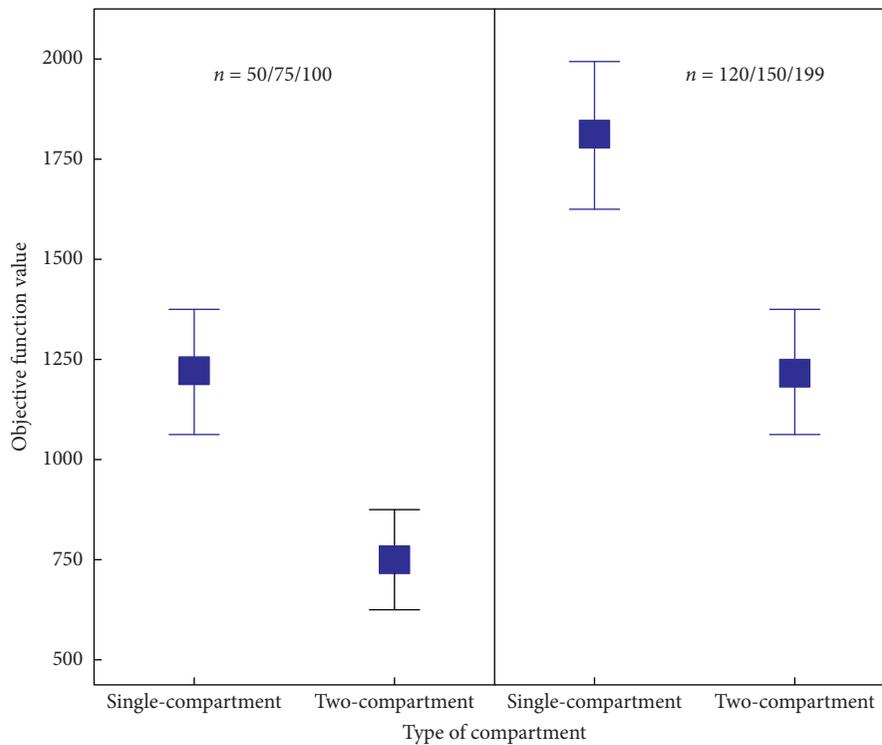


FIGURE 3: Comparison results of vehicles with different compartments.

mentioning that the total distance could be reduced by up to 70.22%, while two-compartment vehicles are selected. The statistical results clearly display the profit of using multi-compartment vehicles.

5. Conclusion

The goal of this paper is to propose an alternative solution for the NP-hard problem, which is dedicated to minimize the total travelled mileage for the multi-compartment vehicle routing problem (MCVRP). A novel improved hybrid ant colony algorithm (IHACO) is proposed, aiming at promoting the global search to reach different regions of

high-quality solutions. A new probabilistic model which utilizes two probability matrices to learn different useful information of high-quality solutions is developed. Then, a heuristic rule based on the nearest neighbor is used to initialize this probabilistic model so that the algorithm's speed of reaching the promising regions is enhanced. Meanwhile, a new local search is constructed to execute the effective exploitation. The efficiency of this method is reflected in the fact that it does not need to judge the capacity constraint and evaluate the objective function. It only needs to judge whether the two routes cross or not, and then, it can directly carry out the exchange operation between two customers. Moreover, two types of VND combined with the

speed-up strategy and the first move strategy are employed to further perform the deep, narrow but rich search in the good solution space that has been found. Finally, numerical experiment results based on a benchmark dataset display the effectiveness and efficiency of IHACO. Future work is to develop an effective ACO-based algorithm to deal with the multiobjective MCVRP and the fuzzy travel time-based MCVRP.

Data Availability

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

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the National Natural Science Foundation of China under Grant nos. 61963022 and 51665025 and the Applied Basic Research Key Project of Yunnan Province.

References

- [1] F. Cornillier, F. F. Boctor, G. Laporte, and J. Renaud, "An exact algorithm for the petrol station replenishment problem," *Journal of the Operational Research Society*, vol. 59, no. 5, pp. 607–615, 2008.
- [2] F. Cornillier, F. F. Boctor, G. Laporte, and J. Renaud, "A heuristic for the multi-period petrol station replenishment problem," *European Journal of Operational Research*, vol. 191, no. 2, pp. 295–305, 2008.
- [3] F. Cornillier, G. Laporte, F. F. Boctor, and J. Renaud, "The petrol station replenishment problem with time windows," *Computers & Operations Research*, vol. 36, no. 3, pp. 919–935, 2009.
- [4] F. F. Boctor, J. Renaud, and F. Cornillier, "Trip packing in petrol stations replenishment," *Omega*, vol. 39, no. 1, pp. 86–98, 2011.
- [5] F. Cornillier, F. F. Boctor, and J. Renaud, "Heuristics for the multi-depot petrol station replenishment problem with time windows," *European Journal of Operational Research*, vol. 220, no. 2, pp. 361–369, 2012.
- [6] E. E. Halvorsen-Weare and K. Fagerholt, "Routing and scheduling in a liquefied natural gas shipping problem with inventory and berth constraints," *Annals of Operations Research*, vol. 203, no. 1, pp. 167–186, 2013.
- [7] R. Lahyani, L. C. Coelho, M. Khemakhem, G. Laporte, and F. Semet, "A multi-compartment vehicle routing problem arising in the collection of olive oil in Tunisia," *Omega*, vol. 51, pp. 1–10, 2015.
- [8] E. D. Chajakis and M. Guignard, "Scheduling deliveries in vehicles with multiple compartments," *Journal of Global Optimization*, vol. 26, no. 1, pp. 43–78, 2003.
- [9] M. Ostermeier and A. Hübner, "Vehicle selection for a multi-compartment vehicle routing problem," *European Journal of Operational Research*, vol. 269, no. 2, pp. 682–694, 2018.
- [10] S. Martins, M. Ostermeier, P. Amorim, A. Hübner, and B. Almada-Lobo, "Product-oriented time window assignment for a multi-compartment vehicle routing problem," *European Journal of Operational Research*, vol. 276, no. 3, pp. 893–909, 2019.
- [11] A. Huebner and M. Ostermeier, "A multi-compartment vehicle routing problem with loading and unloading costs," *Transportation Science*, vol. 53, no. 1, pp. 282–300, 2017.
- [12] L. Muyldermans and G. Pang, "On the benefits of co-collection: experiments with a multi-compartment vehicle routing algorithm," *European Journal of Operational Research*, vol. 206, no. 1, pp. 93–103, 2010.
- [13] M. Elbek and S. Wöhlk, "A variable neighborhood search for the multi-period collection of recyclable materials," *European Journal of Operational Research*, vol. 249, no. 2, pp. 540–550, 2016.
- [14] E. Mofid-Nakhaee and F. Barzinpour, "A multi-compartment capacitated arc routing problem with intermediate facilities for solid waste collection using hybrid adaptive large neighborhood search and whale algorithm," *Waste Management & Research*, vol. 37, no. 1, pp. 38–47, 2019.
- [15] P. Avella, M. Boccia, and A. Sforza, "Solving a fuel delivery problem by heuristic and exact approaches," *European Journal of Operational Research*, vol. 152, no. 1, pp. 170–179, 2004.
- [16] L. M. Hvattum, K. Fagerholt, and V. A. Armentano, "Tank allocation problems in maritime bulk shipping," *Computers & Operations Research*, vol. 36, no. 11, pp. 3051–3060, 2009.
- [17] M. F. Uzar and B. Catay, "Distribution planning of bulk lubricants at BP Turkey," *Omega*, vol. 40, no. 6, pp. 870–881, 2012.
- [18] X. Li, S. C. H. Leung, and P. Tian, "A multistart adaptive memory-based tabu search algorithm for the heterogeneous fixed fleet open vehicle routing problem," *Expert Systems with Applications*, vol. 39, no. 1, pp. 365–374, 2012.
- [19] T. Henke, M. G. Speranza, and G. Wäscher, "The multi-compartment vehicle routing problem with flexible compartment sizes," *European Journal of Operational Research*, vol. 246, no. 3, pp. 730–743, 2015.
- [20] T. Henke, M. G. Speranza, and G. Wäscher, "A branch-and-cut algorithm for the multi-compartment vehicle routing problem with flexible compartment sizes," *Annals of Operations Research*, vol. 275, no. 2, pp. 321–338, 2019.
- [21] M. Reed, A. Yiannakou, and R. Evering, "An ant colony algorithm for the multi-compartment vehicle routing problem," *Applied Soft Computing*, vol. 15, pp. 169–176, 2014.
- [22] M. M. S. Abdulkader, Y. Gajpal, and T. Y. Elmekawy, "Hybridized ant colony algorithm for the multi compartment vehicle routing problem," *Applied Soft Computing*, vol. 37, pp. 196–203, 2015.
- [23] P. V. Silvestrin and M. Ritt, "An iterated tabu search for the multi-compartment vehicle routing problem," *Computers & Operations Research*, vol. 81, pp. 192–202, 2017.
- [24] I. Kaabachi, H. Yahyaoui, S. Krichen, and A. Dekdouk, "Measuring and evaluating hybrid metaheuristics for solving the multi-compartment vehicle routing problem," *Measurement*, vol. 141, pp. 407–419, 2019.
- [25] Z. Wan, G. Wang, and B. Sun, "A hybrid intelligent algorithm by combining particle swarm optimization with chaos searching technique for solving nonlinear bilevel programming problems," *Swarm and Evolutionary Computation*, vol. 8, pp. 26–32, 2013.
- [26] Q. Chen, H. Shi, and M. Sun, "Echo state network-based backstepping adaptive iterative learning control for strict-feedback systems: an error-tracking approach," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–14, 2019.

- [27] Z. C. Li, B. Qian, R. Hu, L. L. Chang, and J. B. Yang, "An elitist nondominated sorting hybrid algorithm for multi-objective flexible job-shop scheduling problem with sequence-dependent setups," *Knowledge-Based Systems*, vol. 173, no. 1, pp. 83–112, 2019.
- [28] M. Mavrouniotis and S. Yang, "Ant algorithms with immigrants schemes for the dynamic vehicle routing problem," *Information Sciences*, vol. 294, pp. 456–477, 2015.
- [29] M. Schyns, "An ant colony system for responsive dynamic vehicle routing," *European Journal of Operational Research*, vol. 245, no. 3, pp. 704–718, 2015.
- [30] X. Wang, T.-M. Choi, H. Liu, and X. Yue, "Novel ant colony optimization methods for simplifying solution construction in vehicle routing problems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3132–3141, 2016.
- [31] S.-H. Huang, Y.-H. Huang, C. A. Blazquez, and G. Paredes-Belmar, "Application of the ant colony optimization in the resolution of the bridge inspection routing problem," *Applied Soft Computing*, vol. 65, pp. 443–461, 2018.
- [32] J. E. Belf and P. R. McMullen, "Ant colony optimization techniques for the vehicle routing problem," *Advanced Engineering Informatics*, vol. 18, no. 1, pp. 41–48, 2004.
- [33] A. Duarte, N. Mladenovi, J. Sánchez-Oro, and R. Todosijevi, "Variable neighborhood descent," in *Handbook of Heuristics*, R. Martí, P. Panos, and M. Resende, Eds., Springer, Cham, Switzerland, 2016.