Research Article
A Hybrid Ant Colony Optimization for Dynamic Multidepot Vehicle Routing Problem

Haitao Xu, Pan Pu, and Feng Duan

School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China

Correspondence should be addressed to Haitao Xu; xuhaitao@hdu.edu.cn

Received 19 July 2018; Accepted 10 September 2018; Published 25 September 2018

Guo Editor: Ahmed S. Hendy

Copyright © 2018 Haitao Xu 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.

In the real world, the vehicle routing problem (VRP) is dynamic and variable, so dynamic vehicle routing problem (DVRP) has obtained more and more attentions among researchers. Meanwhile, due to actual constraints of service hours and service distances, logistics companies usually build multiple depots to serve a great number of dispersed customers. Thus, the research of dynamic multidepot vehicle routing problem (DMDVRP) is significant and essential. However, it has not attracted much attention. In this paper, firstly, a clustering approach based on the nearest distance is proposed to allocate all customers to the depots. Then a hybrid ant colony optimization (HACO) with mutation operation and local interchange is introduced to optimize vehicle routes. In addition, in order to deal with dynamic problem of DMDVRP quickly, a real-time addition and optimization approach is designed to handle the new customer requests. Finally, the t-test is applied to evaluate the proposed algorithm; meanwhile the relations between degrees of dynamism (dod) and HACO are discussed minutely. Experimental results show that the HACO algorithm is feasible and efficient to solve DMDVRP.

1. Introduction

In the history of VRP, the most original and famous routing problem is Traveling Salesman Problem (TSP) [1]. By transforming the TSP, a great number of different kinds of VRP are designed. Generally, VRP has been classified to the following several variants: Capacitated VRP (CVRP), VRP with time windows (VRPTW), VRP with Pickup and Delivery (VRPPD), multidepot VRP (MDVRP), and so on [2–6]. Most particularly, considering that many logistics companies are desirous to improve service quality and save time, the MDVRP receives more and more attentions.

In VRP models, most of the researchers usually define some basic information concerning customers’ locations and demands, available vehicles, etc., which are entirely known before carrying out service. However, VRP is dynamic in most of the actual situations; in other words, the arrangements of customers are changing gradually over time, although a part of customer requests may be known in advance. In recent years, due to the development of new technologies, such as Global Position System, Communication Technology, Information Technology, some researchers have studied DVRP. However, few researchers focus on DVRP with multiple depots, so this paper will study dynamic multidepot vehicle routing problem (DMDVRP) [7].

Comparatively, the number of research projects on the DMDVRP is fewer. T. C. Su introduced a dynamic vehicle control and scheduling of a multidepot physical distribution system in 1999 [8]. For solving dynamic multidepot pickup and delivery problem, T. Huth and D. C. Mattfeld proposed a framework to anticipate dynamic events by model adaptation in 2006 [9]. In 2008, A. Hadjar and F. Soumis introduced multidepot vehicle scheduling problem with time windows, which is a dynamic window reduction. Y. Kuo and C. C. Wang proposed an insertion heuristic to solve dynamic multidepot vehicle routing problem where the pick-up and delivery requests are both considered in 2014 [10]. Liu Y et al. proposed a method with three steps to solve the multiperiod and multidepot dynamic vehicle routing problem with time windows in 2016 [7].

In general, DMDVRP is a NP-hard problem [11], so some exact algorithms (branch and bound algorithm, linear programming, cutting plane algorithm, etc.) are almost impossible to solve it in finite time. However, modern
heuristic algorithms, such as ant colony optimization (ACO) [12, 13], genetic algorithm (GA) [14–16], and particle swarm optimization (PSO) [17], have the ability to generate high quality solutions, although they may not search the best solution. In these approaches, ACO is widely applied to solve different types of VRP.

The original ACO was put forward by Dorigo in his Ph.D. Thesis [18], which is called ant system. However, it is nonconvergent and easy to fall into local optimization. To improve its weaknesses, many researchers have formed several variants of the ant system, such as the ant colony system, the elite ant system, and the largest ant system [19]. In addition, some new mechanisms are proposed to optimize the algorithm, such as enlarging the degree of random search [13], N-Opt local random search, and designing distributed control [20]. For DMDVRP, the objective of the algorithm is not only to search the optimal solution, but also to track the optimal solution through the information of the previous search space over time. Based on this consideration, a hybrid ACO is proposed to solve DMDVRP.

There are two main contributions in this paper. The first contribution of this paper is to use hybrid ACO algorithm to solve DMDVRP, and the method strives to improve the randomization and avoids falling into local search prematurely. In order to improve ACO, the following modifications of ACO are implemented:

1. Dividing region by the nearest distance cluster
2. Optimizing vehicle routes with mutation operation
3. Improving the solutions with local interchange operation

Hybrid ACO has shown its advantages through a large number of comparative experiments based on data sets of different scales.

The second contribution is that a real-time addition and optimization approach is designed to add new real-time customers to known routes and immediately optimize vehicle routes. The process of dynamic optimization can be accelerated.

The remainder of this paper is organized as follows. In Section 2, we describe DMDVRP model and define the problem. In Section 3, the details of hybrid ACO are shown. A real-time addition and optimization approach is introduced in Section 4. Experimental results are discussed in Section 5. Some conclusions are provided in Section 6.

2. Dynamic Multidepot Vehicle Routing Problem

2.1. The Description of DMDVRP. In the real world, MDVRP is affected by the dynamic environment. With the wide application of information technology and Global Positioning System (GPS) technology, the process of logistics service can be tracked and adjusted in real time. Due to the development of these new technologies, the service mode of plan-execute is replaced by the dynamic execution task [21].

Generally speaking, the dynamic nature of DMVRP mainly reflects the uncertainty of customer requests in the service process. Specifically, the type of request includes the uncertain quantity of goods [21–23] and the uncertain service time [24, 25]. This paper focuses on the change of service time and processing orders dynamically according to customer request time. A DMDVRP example is shown in Figure 1. In Figure I(a), three different depots (red square) are responsible to serve the customers (black dots) which are allocated to them, and black and red lines represent initial routes. In Figure I(b), some new customers (blue triangle) are added to system, and some new routes will be generated in time.

2.2. The Formulation of DMDVRP. The DMDVRP can be formalized as follows. An undirected graph \( G = (V, E) \) is established to describe mathematical model. In this model, \( V = \{V_C, V_D\} \) represents the vertex set and \( E = \{(v_i, v_j) \mid v_i, v_j \in V, i < j\} \) is the edge set. \( V_C = \{v_1, v_2, \ldots, v_n\} \) is the set of customers and \( V_D = \{v_{n+1}, v_{n+2}, \ldots, v_{m+n}\} \) is the set of depots. In \( E \), we get distance matrix \( C = (c_{ij}) \) by calculating the Euclidean distance of customers \( v_i \) and \( v_j \). Every customer \( v_i \) has a demand \( q_i \) and needs to be visited once by only one vehicle. There is also a fleet of \( K \) identical vehicles, each with capacity \( Q \).

In the mathematical formulation that follows, binary variable \( x_{ijk} \) is equal to 1 when vehicle \( k \) visits node \( j \) immediately after node \( i \).

Minimize \[ \sum_{i=1}^{n+m} \sum_{j=1}^{m} \sum_{k=1}^{K} c_{ij} x_{ijk}, \] subject to

\[ \sum_{i=1}^{m} \sum_{j=1}^{m} x_{ijk} = 1 \quad (k=1,2,\ldots,K); \] \[ \sum_{j=1}^{m} x_{ijk} = 1 \quad (j=1,2,\ldots,n); \] \[ \sum_{j=1}^{m} q_i x_{ijk} \leq Q \quad (k=1,2,\ldots,K); \] \[ \sum_{i=n+1}^{m} x_{ijk} \leq 1 \quad (k=1,2,\ldots,K); \] \[ \sum_{j=1}^{n} x_{ijk} \leq 1 \quad (k=1,2,\ldots,K). \]

The objective (1) minimizes the total cost. Constraints (2) and (3) guarantee that each customer is served by exactly one vehicle. Vehicle capacity constraint is found in (4). Finally, constraints (5) and (6) check vehicle availability.

2.3. Measuring Dynamism. In most papers, three popular metrics of degree of dynamism [16], effective degree of dynamism, and effective degree of dynamism with TW [22] are introduced to describe dynamism concretely. In this paper, the metric of degree of dynamism (dod) is selected...

Discrete Dynamics in Nature and Society
3. A Hybrid Ant Colony Algorithm for DMDVRP

3.1. Hybrid Ant Colony Optimization. By reviewing previous algorithms about solving VRP, we find that ant colony algorithm (ACO) is a common and effective algorithm. In this paper, the ant colony algorithm is improved by the nearest distance clustering, mutation operation, and 2-Opt algorithm to obtain better solutions. The flowchart of hybrid ant colony optimization (HACO) for the DMDVRP is shown in Figure 2. The details of HACO will be introduced in the following sections.

3.2. The Nearest Distance Cluster Algorithm. Generally, in order to serve customers preferably, the depots are usually placed at the locations which are as close to customers as possible. Therefore, the nearest distance cluster algorithm is proposed to exactly allocate each customer to its nearest depot. By the nearest distance cluster algorithm, a whole DMDVRP is divided into several little DVRPs; meanwhile the whole service area is divided into a set of little regions. An example of the nearest distance cluster algorithm is in Figure 3; as shown, the black triangles are depots, and the different color points represent that customers are allocated to the nearest depot.

3.3. Generate Initial Solutions. In ACO, each ant represents a vehicle and visits all customers once. The complete routes that ants have passed are initial solutions. The ants will decide to select the next customer by formula (8).

\[
p_{ij}(k) = \begin{cases} \frac{(\tau_{ij})^\alpha \times (\eta_{ij})^\beta}{\sum_{l \notin \text{tabu}_k} (\tau_{il})^\alpha \times (\eta_{il})^\beta} & j \notin \text{tabu}_k \\ 0 & \text{otherwise} \end{cases} \tag{8} \]

In formula (8), \(p_{ij}(k)\) is the probability of selecting \(j\) as the next customer of \(i\). \(\tau_{ij}\) and \(\alpha\) are the pheromone density of edge \((i, j)\) and the relative influence of the pheromone trails, respectively. \(\eta_{ij}\) and \(\beta\) are the visibility of edge \((i, j)\) and the relative influence of the visibility values, respectively. \(\text{tabu}_k\) is a list, which stores the unvisited nodes for the \(k\)th ant [12].

3.4. Optimization Operation

3.4.1. Mutation Operation. Mutation operation is derived from genetic algorithms (GA) primitively [23, 24], but it can be applied to fusing other optimization algorithms. For example, the optimization operation can help ACO...
get further solutions in the period of searching. The most important step of mutation operation is to randomly select a solution and exchange customers by mutation probability $P_m$. Therefore, the operational process may produce new solutions and increase the likelihood of searching a better solution.

Generally, $P_m$ has influence on the performance in the mutation. If $P_m$ is so small it is not easy to produce the new individual. If $P_m$ is so large the algorithm becomes a purely random search algorithm. Therefore, the adaptive method is applied to change the mutation probability with the fitness. At the same time, considering the mutation coefficient $k$, it can also affect the performance of mutation. $F_m$ and $k$ can be determined according to the following formulas.

$$P_m = \begin{cases} \frac{P_{m1} - P_{m2} (f_m - f_{avg})}{f_{max} - f_{avg}}, & f_m \geq f_{avg} \\ P_{m1}, & f_m < f_{avg} \end{cases} \quad (9)$$

$$k = \begin{cases} k_1 + k_2 \frac{f_m - f_{avg}}{f_{max} - f_{avg}}, & f_m \geq f_{avg} \\ k_1 + k_2, & f_m < f_{avg} \end{cases} \quad (10)$$

where $f_{max}$ is the largest fitness value, $f_{avg}$ is average in groups, and $f_m$ is the fit value for mutation. $P_{m1}, P_{m2}, k_1, \text{and } k_2$ are initial variables [25].

In most GA, considering that biological gene mutation is a small probability event, the mutation rate is less than 0.1. However, in view of the aim of exploring greater search space, a mutation operation with a larger mutation rate is introduced to optimize route. Figure 4 is a demo of mutation operation process. The red line route and the black line route are serviced by two different vehicles. By exchanging 4 and 6, two new routes (1-3-2-6-1 and 1-4-5-1) are generated and have more appropriate arrangements.

3.4.2. Local Interchange Operation. In all local optimization algorithms, 2-Opt is a common and classical heuristic proposed by Croes [26] in 1958. The main idea is to try to choose a route and exchange two adjacent locations and calculate the results of new route. This operation may avoid local optimization and obtain a better route.

In this paper, the 2-Opt optimization is applied to improve the route. First, swapping the locations of all possible adjacent customers generates some new potential routes. Then we test each new route to see if this pair of exchanges can improve the quality of the route [27]. Finally, the best route will be selected to replace previous old route. Figure 5 is a demo of 2-Opt operation: (a) is the original route and (b) is the new route of implementing the operation of exchanging 2 and 7.

3.5. Update of Pheromone Information. In ACO, the most important step is the update of ant pheromones, which is key to obtain high quality solution. In order to ensure that each link has the same advantages, the following formula will be implemented to update pheromones:

$$\tau_{ij}^{new} = \rho \times \tau_{ij}^{old} + \sum_{k}^{K} \Delta \tau_{ij}^{k} \quad \rho \in (0, 1) \quad (11)$$

In formula (11), $\tau_{ij}^{new}$ and $\tau_{ij}^{old}$ are the new and initial pheromone concentration of link $(i, j)$, respectively, $\rho$ is a constant and regulates the rate of evaporation, $k$ is the number of all routes, $K$ is the number of the routes in specific solution, and $\Delta \tau_{ij}^{k}$ is the increased pheromone of link $(i, j)$ in route $k$. 

---

**Figure 2:** The flowchart of HACO.

**Figure 3:** An example of the nearest distance cluster algorithm.

---

Discrete Dynamics in Nature and Society
4. A Real-Time Addition and Optimization Approach

In this section, a new real-time addition and optimization approach is applied to dynamically add new customer to known routes; meanwhile it can optimize the new generated route. The procedure of this approach is shown in Figure 6.

4.1. The Strategy of Adding New Customers to Known Routes

In the strategy of adding new customers to known routes, a working day is split into 24 same length time slices, and the customer requests are occurring in each time slice. The distances between each new customer and the serving customers or just finished serving customers are computed and compared, and the new customer will be added to the route which is the nearest to the new customer. If the new customer demands exceed the vehicle capacity, they will be handled by arranging another vehicle, and an extra route will be formed.

An example of this strategy is shown in Figure 7; as shown, some customers (black dots) have been known in advance. Red lines and black lines represent initial designed routes to serve known customers. As time goes on, new customers (blue triangle N1, N2, N3) are added to system, and the distances between N1 and the serving or just finished serving customers (black dots 3, 5, 8) will be computed, respectively; then new customer (N1) is added to the nearest route (0-7-8-N1-9-0). Similarly, the new customers (N2, N3) are added to routes (0-1-2-3-N2-0, 0-4-5-N3-6-0), respectively. Although the new generated routes are not the best, the optimized operations will be conducted in the next section.

4.2. Reoptimize Routes Dynamically

Due to adding some new customers to initial routes, the routes need to be reoptimized dynamically. Each new route is divided into two subroutes, and the first subroute which has been served remains unchanged; meanwhile, the remaining part is optimized by mutation and 2-Opt in Section 3.4.

5. Experimental Results and Discussions

In this section, the performance of HACO algorithm solving for DMVVRP is evaluated strictly and completely. Through a series of experiments based on different data sets, the HACO will be analyzed by solution quality, running time, degrees of dynamism, and so on. The data sets are completely open and can be available at http://neo.lcc.uma.es/vrp/vrp-instances/multiple-depot-vrp-instances/. The detailed information of data sets is shown in Table 1. In this table, the numbers of depots, the numbers of customers, and the capacity are explained.

Table 2 shows that the parameters are used in HACO. The number of ants is 30; weight of pheromone is 2; weight of visibility is 1; evaporation rate of pheromone is 0.8; initial mutation rate is 0.8; number of iterations is 100. The algorithm implementation is based on the MATLAB (2010b) language, and the computer configuration is an Intel(R) Core (TM) i5-6500 3.19GHz, 8 GB RAM running Windows 10(x64). All the results are averaged over 10 runs.

5.1. Comparisons Based on Four Different ACO

The comparisons of the solution quality in terms of the route length and running time among four proposed ACO algorithms are implemented in this section, and the four algorithms are ACO, MACO, LACO, and HACO, respectively. The ACO is from Montemanni et al. [28]; MACO and LACO are basic ACO fusing mutation operation and 2-Opt, respectively; modifying ACO with mutation operation and 2-Opt forms the final HACO. In addition, the scale of experimental data sets is between 80 and 288, and the degree of dynamism is 0.3. Table 3 gives the route length and running time of four algorithms. In this paper, the best solutions are in bold entries. From the comparisons, MACO, LACO, and HACO are 3.25%, 0.73%, and 10.9% less than the ACO, respectively. The results indicate that the mutation operation and 2-Opt can improve ACO effectively, and HACO is effective algorithm for solving DMDVVRP.

This result may be attributed to the fact that the introduction of 2-Opt algorithm and mutation operation can improve the randomization and avoid falling into local search prematurely, and make HACO obtain better solutions.
Figure 5: The demo of 2-Opt operation.

Figure 6: The flowchart of real-time addition and optimization approach.

Figure 7: An example of the strategy of adding new customers to known routes.

Table 1: The detailed information of data sets.

| Problem | Depots | Customers | Capacity |
|---------|--------|-----------|----------|
| 01      | 4      | 50        | 80       |
| 02      | 4      | 50        | 160      |
| 03      | 5      | 75        | 140      |
| 04      | 2      | 100       | 100      |
| 05      | 2      | 100       | 200      |
| 06      | 3      | 100       | 100      |
| 07      | 4      | 100       | 100      |
| 08      | 2      | 249       | 500      |
| 09      | 3      | 249       | 500      |
| 10      | 4      | 249       | 500      |
| 11      | 5      | 249       | 500      |
| 12      | 2      | 80        | 60       |
| 13      | 4      | 160       | 60       |
| 14      | 4      | 48        | 200      |
| 15      | 4      | 96        | 195      |
| 16      | 4      | 144       | 190      |
| 17      | 4      | 192       | 185      |
| 18      | 4      | 240       | 180      |
| 19      | 4      | 288       | 175      |

Table 2: HACO parameters settings.

| Parameter | Description                  | Value |
|-----------|------------------------------|-------|
| M         | The number of Ants           | 30    |
| α         | Weight of pheromone          | 2     |
| β         | Weight of visibility         | 1     |
| ρ         | Evaporation rate of pheromone| 0.8   |
| t         | Initial Mutation rate        | 0.1   |
| N         | Number of iterations         | 100   |

In addition, a paired $t$-test is performed to investigate whether there are statistically significant differences between HACO and other improved ACO algorithms. Since it is expected that the solution quality of HACO is better than other ACO, a one-sided alternative hypothesis $H_1$ is given below:

$$H_1 : \mu_{H-ACO} - \mu_{CA} < 0$$ (12)
Table 3: Comparisons based on four different ACO.

| Problem | Length | Time | Length | Time | Length | Time | Length | Time |
|---------|--------|------|--------|------|--------|------|--------|------|
| 01      | 833.06 | 35.27| 839.46 | 52.27| 932.75 | 46.47| 811.72 | 60.45|
| 02      | 716.17 | 26.55| 718.64 | 42.57| 729.37 | 28.33| 599.60 | 46.62|
| 03      | 1061.86| 40.24| 1105.81| 77.84| 1062.03| 43.17| 1033.07| 89.86|
| 04      | 1490.26| 81.59| 1570.71| 154.06| 1541.79| 78.66| 1482.00| 158.92|
| 05      | 1267.82| 52.54| 1323.36| 115.64| 1240.11| 54.99| 1255.38| 130.16|
| 06      | 1313.02| 79.38| 1279.35| 124.45| 1318.54| 63.79| 1274.60| 137.51|
| 07      | 1377.80| 60.93| 1328.30| 109.80| 1322.94| 60.17| 1292.97| 123.60|
| 08      | 8737.04| 147.12| 8264.29| 525.99| 8014.18| 165.17| 7873.47| 599.77|
| 09      | 6723.11| 162.60| 6616.01| 416.62| 6820.88| 168.08| 6279.87| 453.87|
| 10      | 6357.03| 138.60| 6344.92| 343.92| 6103.97| 137.69| 6076.75| 384.96|
| 11      | 6410.59| 146.39| 6262.12| 343.92| 6202.83| 142.02| 6101.29| 366.24|
| 12      | 1539.51| 68.24| 1533.00| 120.76| 1541.37| 69.29| 1524.62| 106.05|
| 13      | 3118.52| 137.43| 2979.31| 242.30| 3122.14| 142.99| 3010.88| 214.50|
| 14      | 1561.51| 23.01| 1399.14| 35.12| 1445.15| 23.61| 1443.02| 39.38|
| 15      | 2383.12| 45.73| 2109.31| 85.79| 2436.77| 44.54| 2003.30| 98.77|
| 16      | 3173.39| 77.47| 3295.31| 148.20| 3469.17| 75.40| 3368.47| 242.12|
| 17      | 3603.98| 105.81| 3724.63| 237.49| 3604.21| 115.32| 3803.42| 165.18|
| 18      | 4035.92| 154.08| 6493.59| 246.79| 6398.13| 121.50| 3570.08| 242.12|
| 19      | 8541.46| 156.46| 4972.95| 446.51| 6468.04| 176.03| 4652.53| 449.91|
| Average | 3381.32| 91.55| 3271.59| 203.64| 3356.65| 92.40| 3011.21| 222.36|

Increased by 0 3.25% 0.73% 10.9% 35.64%

Table 4: Results of paired t-test based on route length.

| Problem | Mean Difference | p value |
|---------|-----------------|---------|
| HACO vs ACO | -370.11 | 0.083 |
| HACO vs MACO | -260.37 | 0.077 |
| HACO vs LACO | -345.44 | 0.041 |

where \( \mu_{HACO} \) and \( \mu_{CA} \) are population mean for HACO and CA, respectively. CA refers to algorithm, which is compared with HACO. For example, if HACO is compared with MACO, the CA will refer to MACO.

Table 4 shows pairs, mean differences for instances, and \( p \) value at statistical level of \( \alpha = 0.1 \). The mean differences of HACO vs ACO, MACO, and LACO are -370.11, -260.37, and -345.44 with \( p \) value of 0.083, 0.077, and 0.041, respectively. This indicates that HACO is statistically significantly different from other ACO; meanwhile HACO fusing mutation and 2-Opt can explore more possibilities and search better solutions.

5.2. Comparisons Based on Different Degrees of Dynamism.

In order to investigate the relation of degrees of dynamism (\( dod \)) and HACO, the experiment based on different \( dod \) (0.3, 0.5, 0.7) is performed in this section. Table 5 gives the detailed routes length and running time of comparisons. In addition, we count the average value, number, and proportion of different \( dod \) in terms of the routes length and running time. When \( dod \) are 0.3, 0.5, and 0.7, the proportions of the best length are 78.95%, 5.26%, and 15.79%, respectively, and the proportions of the best time are 15.79%, 52.63%, and 31.58%, respectively. By the statistical data, the results reveal that \( dod \) is directly proportional to routes length. However, the running time of \( dod \) (0.5 and 0.7) is less than that of \( dod \) (0.3). This illustrates that the proposed real-time addition and optimization approach can reduce algorithm running time, along with the increase of \( dod \).

5.3. The Analyses of Real-Time Addition and Optimization Approach.

In Table 6, the analyses of real-time addition and optimization approach are conducted, and the results are based on \( dod \) (0.3) in this section. In this table, the second, fourth, and fifth columns are the results of static routes length, additional extra routes length, and dynamic routes length, respectively; the third and sixth columns are static and dynamic running time; the seventh and eighth columns are the whole length and running time.

From the analyses of results based on average value, the average dynamic routes length is 2618.7, and it is 23.64% more than the average static routes length. Meanwhile, the average dynamic running time is 24.9 seconds, and it just accounts for 11.20% of the whole running time. These results reveal that the real-time addition and optimization approach can solve the DMDVRP efficiently. The strategy of adding new customers to the nearest known routes is feasible and efficient to solve dynamic problem.
**Table 5:** Comparisons based on different degrees of dynamism.

| Problem | dod = 0.3 | dod = 0.5 | dod = 0.7 |
|---------|-----------|-----------|-----------|
|         | Length    | Time      | Length    | Time      | Length    | Time      |
| 01      | 811.72    | 60.45     | 802.57    | 49.81     | 783.52    | 62.96     |
| 02      | 599.60    | 46.62     | 846.88    | 36.83     | 834.74    | 27.80     |
| 03      | 1033.07   | 89.86     | 1038.82   | 67.68     | 1018.35   | 78.12     |
| 04      | 1482.00   | 158.92    | 1523.84   | 149.00    | 1541.10   | 95.22     |
| 05      | 1255.38   | 130.16    | 1338.78   | 100.67    | 1718.58   | 137.65    |
| 06      | 1292.97   | 123.60    | 1377.39   | 245.13    | 1467.39   | 105.61    |
| 07      | 7873.47   | 599.77    | 7251.37   | 300.56    | 7090.63   | 145.94    |
| 08      | 6279.87   | 453.87    | 6738.20   | 326.09    | 7358.75   | 298.08    |
| 09      | 6076.75   | 384.96    | 6713.71   | 311.21    | 7158.48   | 331.20    |
| 10      | 6101.29   | 366.24    | 6525.99   | 329.49    | 7711.66   | 417.96    |
| 11      | 1524.62   | 106.05    | 1511.59   | 128.06    | 1709.70   | 131.95    |
| 12      | 3100.88   | 214.50    | 3115.21   | 250.86    | 3407.92   | 270.14    |
| 13      | 1443.02   | 39.38     | 1487.92   | 27.40     | 2013.60   | 25.92     |
| 14      | 2003.30   | 98.77     | 2774.69   | 73.84     | 2640.64   | 86.81     |
| 15      | 3236.13   | 165.18    | 3576.19   | 140.96    | 3927.35   | 159.61    |
| 16      | 3368.47   | 242.12    | 4118.43   | 260.63    | 4419.52   | 290.03    |
| 17      | 3803.42   | 357.06    | 4457.53   | 312.86    | 5063.65   | 426.19    |
| 18      | 4652.53   | 449.91    | 5117.38   | 425.45    | 6069.40   | 619.00    |
| Average | 3011.21   | 220.78    | 3247.08   | 191.72    | 3545.00   | 204.92    |
| Count   | 15        | 3         | 10        | 3         | 6         |
| Proportion | 78.95%    | 15.79%    | 5.26%     | 52.63%    | 15.79%    | 31.58%    |

**Table 6:** Results of real-time addition and optimization approach.

| Problem | Len-S | Time-S | Len-E | Len-D | Time-D | All-Length | All-Time |
|---------|-------|--------|-------|-------|--------|------------|----------|
| 01      | 530.36| 50.96  | 152.18| 659.54| 9.49   | 811.72     | 60.45    |
| 02      | 402.51| 40.93  | 67.48 | 532.11| 5.69   | 599.60     | 46.62    |
| 03      | 1787.62| 77.01  | 160.15| 872.92| 12.85  | 1033.07    | 89.86    |
| 04      | 1018.21| 128.09 | 340.43| 1141.57| 30.82  | 1482.00    | 158.92   |
| 05      | 788.61 | 102.83 | 288.92| 966.46| 27.33  | 1255.38    | 130.16   |
| 06      | 880.39| 117.87 | 249.77| 1024.83| 19.64  | 1274.60    | 137.51   |
| 07      | 846.64| 108.18 | 230.12| 1062.85| 15.41  | 1292.97    | 123.60   |
| 08      | 5471.76| 578.84 | 731.18| 7142.29| 20.93  | 7873.47    | 599.77   |
| 09      | 4161.98| 425.35 | 964.16| 5315.71| 28.51  | 6279.87    | 453.87   |
| 10      | 3971.37| 367.37 | 535.12| 5541.63| 17.59  | 6076.75    | 384.96   |
| 11      | 3914.98| 335.17 | 883.43| 5217.86| 31.07  | 6011.29    | 366.24   |
| 12      | 1101.56| 71.93  | 336.47| 1188.15| 34.12  | 1524.62    | 106.05   |
| 13      | 2193.00| 145.49 | 654.49| 2446.39| 69.01  | 3100.88    | 214.50   |
| 14      | 627.68 | 34.95  | 204.10| 1238.92| 4.43   | 1443.02    | 39.38    |
| 15      | 1288.89| 93.90  | 118.85| 1884.45| 4.87   | 2003.30    | 98.77    |
| 16      | 1726.21| 150.22 | 251.35| 2984.78| 14.97  | 3236.13    | 165.18   |
| 17      | 2043.77| 218.86 | 266.67| 3101.80| 23.26  | 3686.47    | 242.12   |
| 18      | 2470.63| 299.59 | 818.56| 2984.85| 57.47  | 3803.42    | 357.06   |
| 19      | 2765.14| 404.28 | 204.28| 4448.24| 45.64  | 4652.53    | 449.91   |
| Average | 1999.54| 197.46 | 392.51| 2618.70| 24.90  | 3011.21    | 222.36   |
Table 7: Comparisons based on other two algorithms.

| Problem | Length | GA | Time | Length | PSO | Time | Length | HACO | Time |
|---------|--------|----|------|--------|-----|------|--------|------|------|
| 01      | 736.39 |    | 61.11| 788.56 | 68.51|      | 811.72 | 60.45|
| 02      | 569.69 |    | 56.96| 598.73 | 60.50|      | 599.60 | 46.62|
| 03      | 931.87 |    | 97.18| 1411.22| 101.09|     | 1033.07| 89.86|
| 04      | 1374.53|    | 126.36| 1475.88| 146.60|     | 1482.00| 158.92|
| 05      | 1396.97|    | 139.64| 1226.40| 144.05|     | 1255.38| 130.16|
| 06      | 1544.30|    | 126.95| 1444.62| 147.58|     | 1443.02| 137.51|
| 07      | 1374.53|    | 126.36| 1411.22| 132.62|     | 1482.00| 158.92|
| 08      | 8633.74|    | 349.46| 6940.90| 387.90|     | 6279.87| 453.87|
| 09      | 7088.22|    | 348.61| 6541.11| 389.81|     | 6076.75| 384.96|
| 10      | 6684.20|    | 353.06| 6686.50| 382.13|     | 6101.29| 366.24|
| 11      | 1616.50|    | 134.80| 1493.42| 129.90|     | 1524.62| 106.05|
| 12      | 4549.78|    | 144.42| 3720.01| 146.57|     | 3100.88| 214.50|
| 13      | 1124.46|    | 35.76 | 1244.96| 41.97 |     | 1443.02| 39.38 |
| 14      | 2305.61|    | 104.37| 2107.53| 111.19|     | 2003.30| 98.77 |
| 15      | 3549.58|    | 151.35| 3398.40| 175.16|     | 3236.13| 165.18|
| 16      | 3876.65|    | 255.01| 3651.31| 274.10|     | 3368.47| 242.12|
| 17      | 4270.49|    | 371.66| 4058.70| 402.31|     | 3803.42| 357.06|
| 18      | 5268.36|    | 364.75| 5486.40| 396.60|     | 4652.53| 449.91|
| Average | 3488.01|    | 199.77| 3219.64| 218.42|     | 3011.21| 222.36|
| Count   | 4      |    | 3     | 12     |      |      |        |      |
| Proportion | 21.05%|    | 15.79%|        |      |      |        | 63.16%|

5.4. Comparisons Based on Other Two Algorithms. In order to better evaluate the performance of the proposed HACO in terms of solution quality in minimizing travel distance, HACO is compared with previously published algorithms. These algorithms are Wang et al.’s [29] genetic algorithm (GA) and Yao et al.’s [30] particle swarm optimization (PSO), respectively.

Table 7 gives the best solution length and running time of this comparison and counts the number and proportion of the best length of each algorithm. From Table 6, the proposed HACO based on DMDVRP finds 12 new best solutions in the 19 problems, accounting for 63.16% of the best, while GA and PSO reach 4 and 3 best possible solutions, accounting for 21.05% and 15.79%, respectively. The results indicate that the HACO is effective algorithm for solving DMDVRP.

6. Conclusions

In the past few decades, few researchers focused on dynamic multidepot vehicle routing problem (DMDVRP). However, DMDVRP has wide application scenarios, so we have a research on DMDVRP. In order to improve the efficiency of ACO, we proposed a hybrid ant colony optimization algorithm (HACO) to solve DMDVRP. The HACO is based on fusion of ACO and the nearest distance cluster, meanwhile using mutation and 2-Opt to optimize routes further. In addition, a real-time addition and optimization approach is proposed to add customers and optimize routes dynamically. In the experiments, the ACO, MACO, LACO, and HACO are compared directly. In order to demonstrate the efficiency of proposed algorithm, the t-test is applied to perform statistical analysis. But beyond that, the relations of HACO and degree of dynamism (dod) are analyzed by a number of tests based on different dod. With the aim of testing the performance of real-time addition and optimization approach, the comparisons of static routes quality and dynamic routes quality are conducted. Experimental results show that the HACO algorithm is feasible and efficient to solve DMDVRP.

Data Availability

The initial data used to support the findings of this study have been deposited in the NEO repository (http://neo.lcc.uma.es/vrp/vrp-instances/). The data sets are valid and can be accessed.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Chinese National Natural Science Foundation (61572165) and Public Projects of Zhejiang Province (LGF18F030006).
References

[1] S. Lin and B. W. Kernighan, “An effective heuristic algorithm for the traveling-salesman problem,” *Operations Research*, vol. 21, pp. 498–516, 1973.

[2] J. W. Olhmann and B. W. Thomas, “A compressed-annealing heuristic for the traveling salesman problem with time windows,” *INFORMS Journal on Computing*, vol. 19, no. 1, pp. 80–90, 2007.

[3] Z. Wang and C. Zhou, “A Three-Stage Saving-Based Heuristic for Vehicle Routing Problem with Time Windows and Stochastic Travel Times,” *Discrete Dynamics in Nature and Society*, vol. 2016, Article ID 7841297, 2016.

[4] M. A. Cruz-Chávez and A. Martínez-Oropeza, “Feasible Initial Population with Genetic Diversity for a Population-Based Algorithm Applied to the Vehicle Routing Problem with Time Windows,” *Mathematical Problems in Engineering*, vol. 2016, Article ID 3851520, 11 pages, 2016.

[5] C. Prodhon and C. Prins, *Metaheuristics for Vehicle Routing Problems*, Springer International Publishing, 2016.

[6] J. Caceres-Cruz, P. Arias, D. Guimarans, D. Riera, and A. A. Juan, “Rich vehicle routing problem: Survey,” *ACM Computing Surveys*, vol. 47, no. 2, article no. 32, 2014.

[7] Y. Liu, I. H. Khalifa, and A. E. Kamel, “The multi-period and multi-depot dynamic vehicle routing problem with time windows,” in *Proceedings of the 3rd IEEE International Conference on Logistics Operations Management, GOL 2016*, pp. 1–6, May 2016.

[8] C.-T. Su, “Dynamic vehicle control and scheduling of a multi-depot physical distribution system,” *Integrated Manufacturing Systems*, vol. 10, no. 1, pp. 56–65, 1999.

[9] T. Huth and D. C. Mattfeld, “Dynamics of the Multi-Depot Pickup and Delivery Problem,” 2007.

[10] Y. Kuo and C. C. Wang, “Using Insertion Heuristic to Solve Dynamic Multi-Depot Vehicle Routing Problem,” *Journal of Algorithms & Optimization*, 2014.

[11] P. Toth and D. Vigo, *An Overview of Vehicle Routing Problems*, Society for Industrial and Applied Mathematics, 2002.

[12] M. Yuan, Z. Yang, and B. Yao, “An improved ant colony optimization for vehicle routing problem,” *European Journal of Operational Research*, vol. 196, no. 1, pp. 171–176, 2009.

[13] C.-H. Chen and C.-J. Ting, “An improved ant colony system algorithm for the vehicle routing problem,” *Journal of the Chinese Institute of Engineers*, vol. 23, no. 2, pp. 115–126, 2006.

[14] F. T. Hanshar and B. M. Ombuki-Berman, “Dynamic vehicle routing using genetic algorithms,” *Applied Intelligence*, vol. 27, no. 1, pp. 89–99, 2007.

[15] A. Cheng and D. Yu, “Genetic algorithm for vehicle routing problem,” in *Proceedings of the 4th International Conference on Transportation Engineering, ICTE 2013*, pp. 2876–2881, Chengdu, China, October 2013.

[16] A. M. F. M. AbdAllah, D. L. Essam, and R. A. Sarker, “On solving periodic re-optimization dynamic vehicle routing problems,” *Applied Soft Computing*, vol. 55, pp. 1–12, 2017.

[17] M. Okulewicz and J. Mandziuk, “Application of Particle Swarm Optimization Algorithm to Dynamic Vehicle Routing Problem,” in *International Conference on Artificial Intelligence and Soft Computing*, pp. 547–558, 2013.

[18] M. Dorigo, V. Maniezzo, and A. Colomi, “Ant system: optimization by a colony of cooperating agents,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.
