Hindawi Discrete Dynamics in Nature and Society Volume 2018, Article ID 1295485, 13 pages https://doi.org/10.1155/2018/1295485



Research Article

Dynamic Vehicle Routing Problems with Enhanced Ant Colony Optimization

Haitao Xu D, Pan Pu D, 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 20 October 2017; Revised 17 January 2018; Accepted 21 January 2018; Published 15 February 2018

Academic Editor: Gabriella Bretti

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.

As we all know, there are a great number of optimization problems in the world. One of the relatively complicated and high-level problems is the vehicle routing problem (VRP). Dynamic vehicle routing problem (DVRP) is a major variant of VRP, and it is closer to real logistic scene. In DVRP, the customers' demands appear with time, and the unserved customers' points must be updated and rearranged while carrying out the programming paths. Owing to the complexity and significance of the problem, DVRP applications have grabbed the attention of researchers in the past two decades. In this paper, we have two main contributions to solving DVRP. Firstly, DVRP is solved with enhanced Ant Colony Optimization (E-ACO), which is the traditional Ant Colony Optimization (ACO) fusing improved *K*-means and crossover operation. *K*-means can divide the region with the most reasonable distance, while ACO using crossover is applied to extend search space and avoid falling into local optimum prematurely. Secondly, several new evaluation benchmarks are proposed, which can objectively and comprehensively estimate the proposed method. In the experiment, the results for different scale problems are compared to those of previously published papers. Experimental results show that the algorithm is feasible and efficient.

1. Introduction

In the past few decades, because of the global developments in transportation and logistics, our lives have been significantly changed. For any local products that need to be sold to other cities or countries, the cost of transportation and logistics is indispensable. Actually, recent research data has shown that the cost of transportation and logistics usually accounts for 20% of the product's value or more [1]; the logistics system has played an ever-growing and indispensable role in daily economic lives. Nevertheless, it also brings many negative effects, for example, air contamination, noises, and traffic accidents [2].

Although transportation and logistics have inevitable consequences for daily lives, efficient vehicles routing arrangements based on optimization algorithm could reduce negative impacts as little as possible, as well as enterprise logistics cost. This is because shortening vehicles running distances will promote the efficiency of vehicles and drivers. In addition, the algorithm can improve customers service quality, reduce exhaust emissions, and promote vehicles dispatch

efficiency [3]. Consequently, the research of vehicle routing problem, which is a significate topic, has grabbed the scholars' attentions during the past few decades [4].

In the history of VRP, the simplest and most famous routing problem is the Travelling Salesman Problem (TSP): given a set of urban locations, a salesman must go to every city once and return to the initial starting city, to find out the shortest travelling routes [5]. By the variant of TSP, researchers design many kinds of VRP; the basic VRP involves a set of customers (each customer should just be serviced once by one vehicle), who need to be serviced by a fleet of vehicles, and all vehicles start and return to the same depot. In addition, due to the limits of the vehicle running length and/or travelling time, service process may need multiple different routes [6, 7]. Actually, the VRP has been classified to several variants, such as Capacitated VRP (CVRP), Multidepot VRP (MDVRP), and VRP with time windows (VRPTW) [8–12].

In most studies of VRP, researchers almost define some basic information concerning customers' locations and demands, available vehicles, and so on, which are entirely known before carrying out service. However, in actual service processes, VRP is dynamic; that is to say, customers' demands and arrangements are changing gradually over time, although a part of customers' demands may be known in advance before starting service. In addition, the DVRP is NP-hard problem, so traditional exact algorithms (linear programming, dynamic programming, greedy algorithm, etc.) are notoriously difficult to solve it under time limitations. However, modern optimization techniques (Ant Colony Optimization (ACO) [13, 14], genetic algorithm (GA) [15–17], particle swarm optimization (PSO) [18], etc.) which have the ability to generate high-quality solutions (although they are not exact) are the most suitable methods to solve DVRP. In these approaches, ACO is a classical and efficient heuristic algorithm.

ACO is a classical bionic algorithm, which is inspired by the process of observing foraging behavior of ant colony. Ant individuals communicate and exchange information by secreting pheromone (a special chemical substance) in the environment. Via sensing concentration of pheromone, ants can choose the appropriate path to reach food sources. This behavior has grabbed people's attentions and created artificial ant systems to resolve combinatorial optimization problems [19].

The initial ACO was proposed by Dorigo in 1991, called ant system. However, it suffered from nonconvergence and local optima problems. A large number of variants of ant system were introduced to make up for its disadvantage effectively, such as elitist ant system, max-min ant system, and ant colony system [20]. Moreover, several novel mechanisms are proposed to promote the performance of algorithm, such as changing rules to enlarge the space of random search [14], N-Opt local random searches, and applying social insects to design distributed control [21]. For the DVRP, the goal of algorithm is not only to search optimum solution, but also to track the optimal solution over time by information of the previous search space. The algorithm needs to be sufficiently quick and flexible to adapt to the changed information. Based on this consideration, the adaptability of algorithm should be enhanced adequately.

ACO is a typical adaptive algorithm since it can transfer information from past environment to new environment and quickly adapt to dynamic changes. In addition, ACO has strong robustness and handles extreme conditions reasonably. In order to better meet dynamic environment, a great number of strategies are introduced to enhance ACO for resolving the DVRP. These can be summarized as (a) maintaining diversity by immigrant schemes [22], (b) memory-based methods [23], (c) multiple population approaches [24], and (d) clustering based algorithms [25].

In this paper, we design an enhanced ACO to solve different scale DVRP. A large number of actual instances show that ACO algorithms can efficiently solve optimization problems in different fields, including the Feature Subset Selection [26], Set Covering Problem [27], and Wireless Sensor Networks [28].

There are two main contributions in this paper. The first is that this paper solves DVRP by enhanced ACO which tries best to improve the degree of randomization and avoid falling

into local search prematurely. In order to enhance the ACO, this paper proposes the following modifications:

- (1) Dividing region by improved *K*-means
- (2) Optimizing the initial solutions with the crossover
- (3) Improving the solutions with 2-Opt.

By a mass of comparative experiments based on different scale data sets, enhanced ACO has shown its advantages.

The second contribution is to design a more equitable evaluation system for DVRP. To date, in most published papers, time-based assessment strategy and cost-based assessment strategy are widespread among VRP. However, those evaluation approaches are biased: they just show the separate cost of several methods. Therefore, except the customary evaluations, the concepts of dynamic degree, vehicles utilization rate, and the t-test are added to the evaluation system.

The reminder of this paper is organized as follows. In Section 2, we describe DVRP model and define the problem. In Section 3, the details of enhanced ACO are shown. The experimental details and results are discussed in Section 4. Some conclusions and future works are provided in Section 5.

2. Problem Description and Definition

In this section, the DVRP will be described in detail. The problem model is defined in the following part.

2.1. Static Vehicle Routing Problem. Generally, static VRP can be defined nearly as follows: to search a route or several routes that link depot with a crowd of customers; meanwhile the total cost is as small as possible.

In the past decades, most papers use an undirected graph G = (V, E) to establish a mathematical model. In the model, $V = \{v_0, v_1, \ldots, v_n\}$ represents the vertex set and $E = \{(v_i, v_j) \mid v_i, v_j \in V, i < j\}$ is an edge set. A set of m homogenous vehicles (having the same and invariable capacity Q) depart from a single depot, which is represented by the vertex v_0 , and must visit total customers that are represented by n vertexes $\{v_1, \ldots, v_n\}$. In E, we calculate the distance of customers v_i and v_j and get distance matrix $C = (c_{ij})$. Every customer v_i has a demand q_i and needs to be visited once by only one vehicle. V is divided into m routes $\{R_1, \ldots, R_m\}$ that include all customers. The distance of route $R_i = \{v_0, v_1, \ldots, v_{k+1}\}$ where $v_k \in V$ and the depot $v_{k+1} = v_0$ is calculated by

Cost
$$(R_i) = \sum_{j=0}^{k} c_{j,j+1},$$
 (1)

and calculating cumulative $Cost(R_i)$ to total cost of solutions S is as follows:

$$f_{\text{VRP}}(S) = \sum_{i=1}^{m} \text{Cost}(R_i).$$
 (2)

On the whole, the static VRP needs to observe the following constraints [15]:

- (1) Each vehicle starts from and returns to the same depot.
- (2) Each customer is to be visited once by only one vehicle exactly.
- (3) Each vehicle (assuming all vehicles are of the same model) has a capacity limitation.
- (4) The vehicle arrives and service time must satisfy the stipulated time.

2.2. Dynamic Vehicle Routing Problem

2.2.1. A General Definition. In the real world, the VRP is subject to dynamic environments. With the maturity of Global Positioning System (GPS) and widespread use of smart phones, tracing and managing a fleet of vehicles in real time can be realized easily. Due to the developments of new technologies, the process of plan-execute is replaced by dynamic planning vehicle routes [29].

Generally, the dynamism has mainly revealed the uncertainty of customer requests during the services. More concretely, the varieties of requests can be the number of goods [30–32] and services [33]. The travel time [34] and service time, two dynamic factors of the most read-world environments, have been taken into account. In this paper, dynamism focuses on the changes of service time. According to customer request time, the algorithm handles the orders dynamically.

2.2.2. DVRP Model. We study a classical DVRP model which is proposed by Montemanni et al. [35]. In this paper, DVRP is regarded as a variety of the ordinary static VRP by dividing a whole DVRP into a set of standard VRP and then solving them in sequence with ACO. A number of vehicles have been arranged to serve customers that are known in advance; meanwhile many new customers' demands are emerging constantly over time. These newly joined customers' demands should be sent to the vehicles that are working or are handled by additional vehicles according to new customers' required time. Thus, there are always some customers which have been serviced and new customers who wait to be serviced at any moment in working day. If a day is divided into a lot of little time periods, the DVRP can be regarded as a set of standard static VRP in every time period. Due to the fact that VRP is a NP-hard problem, this indicates that the DVRP also is a NPhard problem [36], so DVRP must be also handled in each setting time period. A DVRP example is shown in Figure 1; as shown, some known customers (black dots) orders have been known in advance. Red lines and black lines represent initial designed routes to service known customers. As time goes on, new customer/s (blue triangle) orders are added to the system; thus the additional new customers are inserted into existing routes and will generate more new routes [15].

2.2.3. Measuring Dynamism. In different problems, the levers of dynamism cannot be the same. The dynamism is usually characterized by the frequency of changes and the urgency of requests [37]. Three metrics have been applied to describe dynamism concretely; they are, respectively, degree

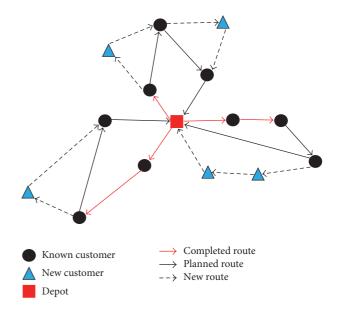


FIGURE 1: A three-vehicle DVRP with 5 unknown customer orders.

of dynamism [17], effective degree of dynamism, and effective degree of dynamism with TW [38].

This paper adopts the DVRP model in [35] and regards DVRP as a set of static VRP. Therefore, this paper selects the metric, *degree of dynamism* (dod), which is the ratio of the known to unknown customers before the system starts to serve:

$$dod = \frac{count \ of \ customers \ (known \ in \ advance)}{all \ customers}, \quad (3)$$

where dod \in [0, 1]. If dod is 1, all customers are known in advance and the problem is completely static, while if it is 0, no customers are known in advance and the problem is completely dynamic [17].

2.3. Converting DVRP to Static VRP with Event Scheduler System. The event scheduler system is aimed to manage customers' orders, including accepting orders, distributing orders, and creating static problems. It is used as a dispatching center that connects new real-time orders with the optimization procedure. Firstly, the system needs to submit the known customers' orders to optimization procedure, let it create static problems, and serve the known customers; meanwhile the system accepts new real-time customers' orders. Secondly, during the process of handling the static problem, if new added orders need to be dealt with in time, the new orders should be added in prior unserved orders list immediately. Lastly, today's remaining orders are arranged to the next working day. An event scheduler system is shown in Figure 2. As shown, the system accepts the customers' orders, creates static problems, and sends problems to the optimization procedure (our algorithm). After the enhanced ACO (E-ACO) produces static solutions and returns to the system, then system commits orders.

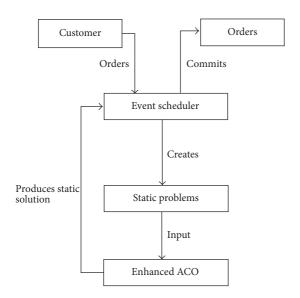


FIGURE 2: The flow chart of event scheduler.

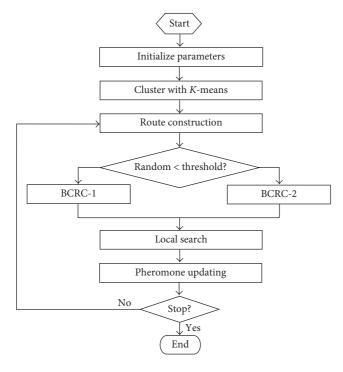


FIGURE 3: The flow chart of E-ACO.

3. Enhanced ACO for DVRP

3.1. Enhanced Ant Colony Optimization. By reviewing previous papers about Ant Colony Optimization, we discover that ACO is an efficient algorithm for solving VRP. For example, Yu and Yang solved the period vehicle routing problem with time windows (VRPTW) by adopting ACO [39]. In this paper, in order to gain better solutions, K-means, crossover, and 2-Opt are applied to enhance ACO. The flowchart of E-ACO for the DVRP is shown in Figure 3. The following sections will introduce the details of E-ACO.

3.2. The Improved K-Means. K-means is a classical clustering algorithm, and its objective function is the sum of distances between clustering center and data points. The main idea of K-means is to divide a set of n points into k regions according to minimizing objective function and let each region be compact as far as possible.

In K-means algorithm, the metric is the Euclidean distance. The sum of error squares J_c is calculated by (4), and it is applied to classify the initial clustering centers [25]. In (4), X_i is a series of clustering centers, m_i denotes the average value of the clustering center i, and p are the data points included in i

$$J_c = \sum_{i=1}^k \sum_{p \in X_i} \|p - m_i\|^2.$$
 (4)

Although the traditional K-means algorithm can deal with clustering problems, it relies on k value. In this paper, (5) is applied to determine the k value. In (5), m denotes the average value of all data points, and other parameters are the same as (5). Once the minimal F is obtained, the k value will be determined ultimately [40]

$$F = \sum_{i=1}^{k} |m_i - m| + \sum_{i=1}^{k} \sum_{p \in X_i} |p - m_i|.$$
 (5)

According to the modified K-means, the data points are divided into k different regions reasonably; then E-ACO will handle each region, respectively.

3.3. Generating Initial Solutions. In ACO, the scale of colony is defined as *P*, an individual ant represents a vehicle, and the route is generated by gradually visiting customers until all customers have been visited. The customers, who have been visited or who violated the vehicle capacity constraint, are stored in the tabu list, and the list can ensure that this ant does not select those customers again.

The strategy of how to decide to select the next visiting customer depends on a probabilistic rule, which takes into consideration the visibility of the ants and the pheromone information. Therefore, the ants will rely on the following formula to decide the next customer j for the kth ant at the ith node

$$p_{ij}(k) = \begin{cases} \frac{\left(\tau_{(i,j)}\right)^{\alpha} \times \left(\eta_{(i,j)}\right)^{\beta}}{\sum_{I \notin \text{tabu}} \left(\tau_{(i,I)}\right)^{\alpha} \times \left(\eta_{(i,I)}\right)^{\beta}} & j \notin \text{tabu}_{k} \\ 0 & \text{otherwise,} \end{cases}$$
(6)

where $p_{ij}(k)$ is the probability of selecting j as the next customer of i on the route, $\tau_{(i,j)}$ is the pheromone density of edge (i, j), $\eta_{(i,j)}$ is the visibility of edge (i, j), α and β are the relative influence of the pheromone trails and the visibility values, respectively, and tabu $_k$ is the set of the unfeasible nodes for the kth ant [13].

3.4. Optimization Operation

3.4.1. Crossover Operation. In general, crossover operation comes from the genetic algorithm (GA) [41], but it can be

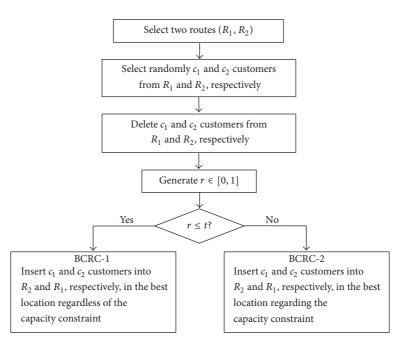


FIGURE 4: The modified BCRC.

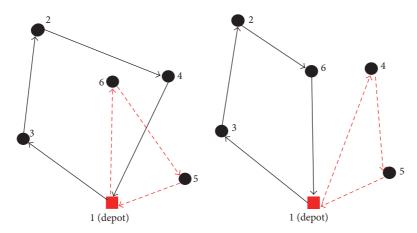


FIGURE 5: The example of crossover operation.

applied to other algorithms. For example, the operator can help the ACO reach further solutions in the search space. Crossover operation's main idea is to randomly select two tours and exchange, respectively, the customers of their solution by crossover rate. Consequently, the process of operation will probably generate new solutions and increase the possibility of finding better solutions.

This paper modifies the version of the Best Cost Route Crossover (BCRC) in [42]. The detailed steps concerning the conventional BCRC are introduced in [39, 42]; the new version that we modify is shown in Figure 4, where r is randomly generated decimal; it is used to compare with the crossover threshold (t). The best insertion location is a point of selected route which is exchanged with another selected route and makes the new routes' results minimum; the results can be calculated by (1). In this approach, the customer (c) is

inserted into primary customers (a, b), and the newest cost is calculated as follows:

$$Cost_{c,ab} = Dist (c,a) + Dist (c,b) - Dist (a,b).$$
 (7)

In order to implement this idea, the E-ACO system initially sets t=1 (normal BCRC). If it obtains the best route in ten continuous tests, it is added to the best list. When the system is aware of the fact that this has been in the best list, the system will be far from the list and reach a global optimum; the following threshold will be decreased by 10% and the system will run ten continuous tests again. With time, the threshold is less than 0.1; in that condition it is reset to 1.0 again [17, 43].

Finally, a best strategy based on different crossover thresholds is selected, and it ensures that the best solution is detected over time. Figure 5 is an example of crossover

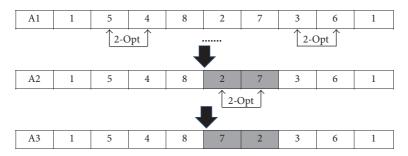


FIGURE 6: The example of 2-Opt operation.

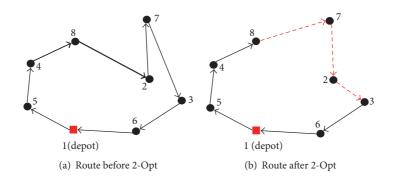


FIGURE 7: The demo of 2-Opt operation.

operator process. The black line route (1-3-2-4-1) and the red line route (1-6-5) are two different routes. According to exchanging 4 and 6, two new routes (1-3-2-6-1 and 1-4-5-1) are structured.

3.4.2. Local Search. 2-Opt is a classical local search heuristic, which was proposed by Croes [44] in 1958. The main idea is to select a route and exchange the two neighboring locations. This operation provides a way to obtain better solution and avoid the algorithm falling into local optimum.

In this paper, the operation of 2-Opt is applied to optimize routes. Firstly, exchanging all possible neighboring customers' locations generates some new routes. Then each new route is tested to see whether this pair exchange can improve the solutions' quality [45]. Finally, the best solutions will be adopted. The operation has been applied to several ACO applications (Chen and Ting, 2006 [14] and Gao et al. 2016 [25]) for the VRP.

Figure 6 is an example of implementing. In this figure, A1, A2, and A3 denote three routes. In A1, firstly, customers 5 and 6 exchange locations and obtain a new route. Then exchanging 4 and 8 forms another route, and so on, to exchange 3 and 6. Finally, calculate each new route's value by (1), and find the minimal value of all new routes. By comparing all new routes, exchanging 2 and 7 (A2) will become the best route, so exchanging 2 and 7 forms route A3. Figure 7 is the graphic of this operation; the elements are the same as those in Figure 1. Figure 7(a) is the route before implementing the operation of exchanging 2 and 7, and Figure 7(b) is the route of implementing 2-Opt.

3.5. Update of Pheromone Information. One of the most important steps of ACO is the updating ant pheromone. It is the key to adapting the self-learning technique of ACO and obtaining high-quality solutions. In order to simulate the process of pheromone evaporation, reducing pheromone concentration is applied to all links, and it can ensure that no link has unique advantages. The approach is implemented with the following updating formula:

$$\tau_{ij}^{\text{new}} = \rho \times \tau_{ij}^{\text{old}} + \sum_{k}^{K} \Delta \tau_{ij}^{k} \quad \rho \in (0, 1),$$
 (8)

where τ_{ij}^{new} is the final pheromone of link (i,j), τ_{ij}^{old} is the initial pheromone of link (i,j), ρ is a constant that adjusts the speed of evaporation, k is the number of all routes, K is the number of the routes in the solution, and $\Delta \tau_{ij}^k$ is the increased pheromone of link (i,j) in route k.

In this paper, the rule of updating pheromone refers to the ant-weight strategy put forward by Yang et al. [46]. The strategy is indicated as

$$\Delta \tau_{ij}^{k} = \begin{cases} \frac{Q}{K \times L} \times \frac{D^{k} - d_{ij}}{m^{k} \times D^{k}} & \text{if link } (i, j) \text{ on the } k \text{th route} \\ 0 & \text{otherwise,} \end{cases}$$
(9)

where *Q* is a constant and *L* is the sum of all routes' lengths, that is, $L = \sum_k D^k$, D^k is the *k*th route length, d_{ij} is the length

TABLE 1: E-ACO 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 crossover threshold	1
N	Number of iterations	100-500

of edge (i, j), and m^k $(m^k > 0)$ is customers' number in the kth route

The ant-weight strategy is based on actual solution quality to update the increased quantity of pheromone, which includes two parts: the local pheromone increment and the global pheromone increment. In this strategy, the increased quantity of global pheromone $Q/(K \times L)$ is relevant to the total paths length, while the local pheromone increment $(D^k - d_{ij})/(m^k \times D^k)$ is related to the corresponding link (i, j) contribution to this solution.

In addition, in order to avoid the risk getting in local optimization, upper and lower limits $[\tau_{\min}, \tau_{\max}]$ are set as follows:

$$\tau_{\min} = \frac{Q}{\sum_{i} 2d_{0i}},$$

$$\tau_{\max} = \frac{Q}{\sum_{i} d_{0i}},$$
(10)

where d_{0i} is the distance between the depot and the *i*th customer [13].

4. Experimental Results and Discussions

In this section, the performance of the enhanced Ant Colony Optimization algorithm to solve DVRP will be assessed rigorously and integrally. By a number of experiments based on different scales of data sets, we analyze the performance of the proposed algorithm by solution quality, utilization rate of vehicles, and degrees of dynamism. The data sets are open and available at http://neo.lcc.uma.es/vrp/.

The E-ACO parameters used for instances are shown in Table 1. The algorithm is implemented by the MATLAB (version: R2010b) language, and the configuration of our experimental computer is an Intel® Core™ i5-6500 3.19 GHz, 8 GB RAM running Windows 10 (x64). All the results are averaged over 25 runs.

4.1. Comparison Based on Small and Medium Scale of Data Sets. In this part, a comparison of the solution quality in terms of the best value and the average value among three proposed ACO algorithms is implemented; the three algorithms are ACO, K-ACO, and E-ACO, respectively. The ACO is from Montemanni et al.'s [35] ACO, the K-ACO is basic ACO fusing K-means and 2-Opt, and the modifying K-ACO with crossover operation forms the final E-ACO. In addition, the scale of data sets is between 50 and 199.

Table 2 gives the best and average value of three approaches. The numbers in bold are the best results among three algorithms. It can be observed that E-ACO attains 17 out of 20 best values and 12 out of 20 average values compared with ACO and K-ACO. This may be attributed to the fact that the introduction of *K*-means algorithm can divide the search space reasonably and make the ACO using crossover operation obtain better solutions in each clustering region.

Meantime, we perform statistical analysis using a paired t-test to investigate whether there are statistically significant differences between E-ACO and other algorithms according to the solution quality. Since it is expected that the best solution of E-ACO is better than other approaches, a null hypothesis, H_0 , is given below:

$$H_0: \mu_{\text{E-ACO}} - \mu_{\text{CA}} < 0,$$
 (11)

where $\mu_{\text{E-ACO}}$ and μ_{CA} are population mean for E-ACO and CA, respectively. CA refers to the compared algorithms; if E-ACO is compared with K-ACO, the CA will be K-ACO.

Table 3 shows pairs, mean differences for instances, and p value at statistical level of $\alpha = 0.05$. In Table 3, the union of group 1 and group 2 shows the two improved ACO (K-ACO and E-ACO) compared with the original ACO. By observing the p value of group 1 and group 2 among best and average values, the mean differences of ACO versus K-ACO are 69.79 and 178.55 with p value of 0.834 and 0.699, respectively; this indicates that the basic ACO using *K*-means and 2-Opt is worse than ACO in [35]. However, the mean differences of ACO versus E-ACO are -136.51 and -38.88 with p value of 0.890 and 0.972, respectively; E-ACO is statistically significantly different from ACO. This indicates that K-ACO fusing crossover can be able to explore more possibilities and search for better solutions. The union of group 2 and group 3 shows the mean of differences and p value; the analysis is similar to group 1 and group 2; this analysis tests the efficiency of E-ACO again.

Although it seems that *K*-means did not contribute to the E-ACO, the efficiency of *K*-means is shown in the section of comparison based on high scale of data sets.

In addition, this paper also compares the utilization rates of vehicles among ACO, K-ACO, and E-ACO. Because Montemanni did not discuss utilization rate of vehicles in [35], the data is blank. The vehicles' utilization rate is a significant value for assessing the performance of the approach; it can show whether each vehicle is utilized fully.

Table 4 shows utilization rate of vehicles. Nine out of 20 results from E-ACO outperform 7 out of 20 from K-ACO, and the other 5 instances are of the same value. Due to application of *K*-means, the utilization rate of vehicles among K-ACO and E-ACO is more than 80%. This indicates that *K*-means is an appropriate approach to improve the utilization rate of vehicles. If any VRP has limited number of vehicles, *K*-means should be preferred.

4.2. Comparison Based on High Scale of Data Sets. Although most of the studies of vehicle routing problem adopt the small and medium scale of data sets, this paper will add the high scale of data sets to our experiment. The data sets' scale is

Problem	A	CO	K-A	ACO	E-A	CO
Problem	Best	Average	Best	Average	Best	Average
f71	311.18	358.69	286.42	315.31	259.71	297.08
c50	631.30	681.86	611. 27	652.45	607.21	647.21
c75	1009.36	1042.39	982.30	1136.90	924.71	1045.44
c100	973.26	1066.16	1021.00	1161.50	973.40	1044.96
c100b	944.23	1023.60	921.41	1000.80	869.22	950.17
c120	1416.45	1525.15	1137.50	1297.30	1108.15	1197.68
c150	1345.73	1455.50	1480.00	1730.90	1378.63	1472.40
c199	1771.04	1844.82	1732.00	1970.60	1561.12	1836.86
tai75a	1843.08	1945.2	1657.10	2083.30	1690.91	1983.92
tai75b	1535.43	1704.06	1541.90	1680.70	1509.56	1647.78
tai75c	1574.98	1653.58	1395.10	1591.60	1329.42	1470.60
tai75d	1472.35	1529.00	1493.70	1663.10	1409.14	1661.73
tai100a	2375.92	2428.38	2497.60	2714.20	2281.70	2550.64
tai100b	2283.97	2347.90	2365.80	2624.70	2255.83	2500.72
tai100c	1562.30	1655.91	1547.60	1764.50	1442.45	1743.07
tai100d	2008.13	2060.72	1958.80	2227.50	1581.36	1843.82
tai150a	3644.78	3840.18	3800.60	4206.40	3307.63	3684.03
tai150b	3166.88	3327.47	3270.20	3654.80	3128.00	3439.38
tai150c	2811.48	3016.14	2795.00	2886.20	2583.36	2729.15
tai150d	3058.87	3203.75	2981.20	3245.21	2808.99	3186.08
average	1787.04	1885.52	1856.83	2064.07	1650.53	1846.64
Count	2	8	1	0	17	12

TABLE 2: Comparison of ACO, K-ACO, and E-ACO.

TABLE 3: *t*-test significance results between each ACO.

	(Group 1	Gr	oup 2	Gr	oup 3
Problem	ACO v	ersus K-ACO	ACO vei	rsus E-ACO	K-ACO ve	ersus E-ACO
	Best	Average	Best	Average	Best	Average
Mean difference	69.79	178.55	-136.51	-38.88	-206.30	-207.43
<i>p</i> value	0.834	0.699	0.809	0.972	0.650	0.663

from 225 to 480. Because we are unable to find previous papers which use these Kelly data sets, the ACO is replaced by the basic ACO, which was not optimized with *K*-means and crossover. K-ACO and E-ACO are the same as those in Section 4.1.

Table 5 gives the best and average value of three approaches. The numbers in bold are the best results among three algorithms. It can be observed that E-ACO attains 6 out of 8 best values and 5 out of 8 average values compared with basic ACO and K-ACO. This may be attributed to the fact that the introduction of *K*-means algorithm can divide the search space reasonably and make the ACO using crossover operation obtain better solutions in each clustering region.

In order to show the efficiency of each optimal operation, we also applied the paired t-test. The definition of paired t-test is the same as Section 4.1. By observing the basic ACO versus K-ACO in Table 6, the mean differences of ACO versus K-ACO are -64.99 and -119.12 with p value of 0.682 and 0.654, respectively; K-ACO is statistically significantly different from basic ACO. This also indicates that the basic ACO using K-means is better than basic ACO and the K-means improved the ACO effectively. The mean differences of K-ACO versus E-ACO are -47.01 and -43.20 with p value of

0.835 and 0.866, respectively; this indicates that crossover operation has a contribution to E-ACO. By comparison based on high scale of data sets, the contributions of *K*-means and crossover to E-ACO are confirmed completely.

4.3. Comparison Based on Different Degrees of Dynamism (dod). In this part, for the sake of evaluating different degrees of dynamism's effect to ACO, we will perform some ACO tests based on different dod (1/3, 1/2, and 2/3). Table 7 gives the best and average results of this comparison. From this table, the same data set with different degrees of dynamism will obtain diverse results. In order to clearly show the results, we count the number and proportion of different dod in terms of the best value and the average value. In different dod, the distribution of the best value and average value is relatively stable and balanced; this illustrates that E-ACO is robust and stable.

4.4. Comparison with Previously Published DVRP Systems. In order to evaluate the performance of the proposed enhanced ACO regarding the solutions quality in terms of minimizing the travelling distances, comparisons have been conducted between it and previously published DVRP systems. These

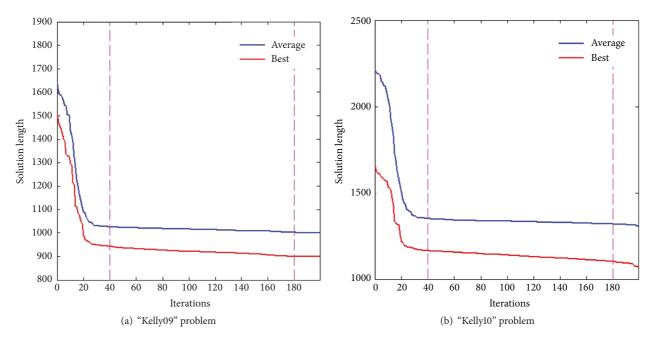


FIGURE 8: Examples of convergence characteristics.

Table 4: Comparison of vehicles' utilization rates.

Problem		Utilization	
Problem	ACO	K-ACO	E-ACO
f71	-	95.70%	95.70%
c50	-	80.94%	69.38%
c75	-	88.31%	88.31%
c100	-	91.13%	81.00%
c100b	-	90.50%	90.50%
c120	-	98.21%	85.94%
c150	-	93.13%	93.13%
c199	-	88.50%	93.71%
tai75a	-	86.54%	95.20%
tai75b	-	88.78%	80.71%
tai75c	-	84.43%	94.99%
tai75d	-	83.43%	92.70%
tai100a	-	83.00%	98.09%
tai100b	-	88.23%	81.45%
tai100c	-	86.03%	93.85%
tai100d	-	80.61%	95.27%
tai150a	-	88.37%	94.26%
tai150b	-	94.91%	88.58%
tai150c	-	87.85%	87.85%
tai150d	-	90.99%	90.99%

DVRP systems are Hanshar and Ombuki-Berman's [15] Tabu Search (TS) and GA, Khouadjia et al.'s [41] Variable Neighborhood Search (VNS) with a local search, and particle swarm optimization (PSO) with a local search, Abdallah's [17] GAbased DVRP.

Table 8 gives the best and average value of this comparison and counts the number and proportion of the best values of

each algorithm. In this paper, the best found solutions are in bold. This table shows that the proposed E-ACO based on DVRP finds 10 new best solutions in the 20 problems, accounting for 50% of the best, while GA-based DVRP reaches 6 best possible solutions, accounting for 30%. The DVRP-GA, DAPSO, and VNS produce number of the best solutions being 2, 1, and 1, respectively.

Meanwhile, we perform statistical analysis using a paired t-test to investigate whether there are statistically significant differences between E-ACO and other algorithms according to the solution quality. Table 9 shows pairs, mean differences for instances, and p value at statistical level of $\alpha = 0.05$. As is shown in Table 9, E-ACO is statistically significantly different from tabu, DVRP-GA, DAPSO, and VNS; their p values are 0.841, 0.899, 0.807, and 0.800 and mean differences are -62.99, -33.17, -65.26, and -67.13, respectively. Particularly, GA-based DVRP is the relatively newer and better results; E-ACO is statistically significant from it with 0.983 and mean difference of -5.45. This analysis indicates that the E-ACO performs as well as other algorithms which are the most effective approaches recently proposed in this paper.

4.5. Convergence Characteristics of the E-ACO. In this section, we investigate the proposed approach by observing its convergence characteristics while solving DVRP. Here we will consider two examples of problem "Kelly09" and problem "Kelly10."

In Figure 8, (a) and (b) show the convergence characteristics. The red line and blue line are the trend of "best value" and "average value" with iterations. In (a), after 30 iterations, E-ACO almost begins to converge. From 30 to 180 iterations, convergence rate is very slow. In Figure 8(b), the status of convergence is similar to (a). Quick convergence of algorithm may harm optimization problem, but it is necessary for DVRP

Problem	Basi	c ACO	K-A	K-ACO		E-ACO	
Problem	Best	Average	Best	Average	Best	Average	
Kelly09	894.08	1034.1	868.62	936.35	830.98	906.71	
Kelly10	1169.5	1380.7	1181	1243.9	1072.5	1244.7	
Kelly11	1580.4	1808.2	1415.4	1618.3	1311.1	1523.6	
Kelly12	1964.4	2311.6	1722.8	2047.5	1755.8	2068.3	
Kelly13	1160.6	1245.5	1108.8	1232.2	1096.0	1188.9	
Kelly14	1411.6	1631.3	1526.5	1649.1	1363.1	1548.2	
Kelly15	1863.7	2077.6	1784.9	1960	1781.4	1861	
Kelly16	2286.8	2593.9	2203.2	2442.6	2224.2	2442.9	
Average	1541.39	1760.36	1476.40	1641.24	1429.39	1598.04	
Count	0	0	2	3	6	5	

TABLE 5: Comparison of basic ACO, K-ACO, and E-ACO.

Table 6: *t*-test significance results between each ACO.

Problem	Basic AC	O versus K-ACO	Basic AC	CO versus E-ACO	K-ACO	versus E-ACO
Tiobiciii	Best	Average	Best	Average	Best	Average
Mean difference	-64.99	-119.12	-112	-162.32	-47.01	-43.20
p value	0.682	0.654	0.906	0.545	0.835	0.866

Table 7: Comparison of solutions based on different degrees of dynamism.

Problem	dod	= 1/3	dod	= 1/2	dod	= 2/3
Problem	Best	Average	Best	Average	Best	Average
f71	284.15	334.14	290.74	337.56	295.82	337.61
c50	611.97	707.99	609.32	702.87	620.12	709.72
c75	1055.60	1137.40	1042.20	1128.40	1102.30	1136.90
c100	1096.90	1263.80	1116.50	1268.90	1124.00	1267.50
c100b	1009.90	1065.60	998.40	1067.10	992.41	1068.80
c120	1153.40	1293.60	1152.00	1303.70	1137.50	1297.30
c150	1574.80	1745.00	1609.00	1742.30	1587.00	1730.90
c199	1955.20	2268.70	2064.70	2261.70	2039.00	2270.60
tai75a	1953.70	2108.00	1895.50	2105.60	1957.10	2083.30
tai75b	1575.20	1677.00	1490.10	1660.70	1541.90	1680.70
tai75c	1411.50	1604.20	1443.10	1609.60	1395.10	1591.60
tai75d	1533.60	1668.20	1418.30	1980.60	1593.70	1663.10
tai100a	2657.80	2944.80	2656.10	2914.00	2597.60	2914.20
tai100b	2367.30	2616.10	2387.80	2619.00	2365.80	2624.70
tai100c	1670.50	1860.60	1678.80	1864.60	1687.60	1864.50
tai100d	1988.10	2229.20	2013.90	2218.50	1958.80	2227.50
tai150a	3755.30	4210.50	3777.10	4220.80	3800.60	4206.40
tai150b	3363.40	3661.50	3273.60	3666.50	3370.20	3654.80
tai150c	2889.40	3198.50	2941.70	3208.80	2995.00	3186.20
tai150d	3348.60	3720.20	3296.00	3752.80	3175.10	3765.00
Count	7	7	6	6	7	7
Prop	35%	35%	30%	30%	35%	35%

to search the solution. For DVRP, the goal of the algorithm is not only to search for the best solution, but also to attain the best solution in short term.

By a large convergence experiment, we found that E-ACO can almost obtain the best value compared with tabu, DAPSO, VNS, and GA-based DVRP in 40 iterations.

5. Conclusions and Future Work

In most published papers, their methods are proposed to solve static VRP. However, in the real world, the vehicle routing problems are dynamic. So we proposed an enhanced Ant Colony Optimization algorithm to solve DVRP; E-ACO

TABLE 8: Comparison between the published systems.

Droblom	T	Tabu	DVR	DVRP-GA	DAPSC	OSo	[\(\)	NNS	GA-based DVRP	d DVRP	E-ACO	00
rioniciii	Best	Average	Best	Average	Best	Average	Best	Average	Best	Average	Best	Average
f71	280.23	306.33	301.79	309.94	279.52	312.35	304.32	325.18	288.3	309.49	259.71	297.08
c50	603.57	627.90	570.89	593.42	575.89	632.38	599.53	653.84	566.01	587.34	607.21	647.21
c75	981.51	1013.82	981.57	1013.45	970.45	1031.76	981.64	1040.00	944.46	990.78	924.71	1045.44
c100	997.15	1047.60	961.10	987.59	988.27	1051.50	1022.92	1087.18	943.89	988.15	973.40	1044.96
c100b	891.42	932.14	881.92	900.94	924.32	964.47	866.71	942.81	869.41	904.03	869.22	950.17
c120	1331.80	1468.12	1303.59	1390.58	1276.88	1457.22	1285.21	1469.24	1288.66	1399.4	1108.15	1197.68
c150	1318.22	1401.06	1348.88	1386.93	1371.08	1470.95	1334.73	1441.37	1273.5	1359.25	1378.63	1472.40
c199	1750.09	1783.43	1654.51	1758.51	1640.40	1818.55	1679.65	1769.95	1646.36	1700.54	1561.12	1836.86
tai75a	1778.52	1883.47	1782.91	1856.66	1816.07	1935.28	1806.81	1954.25	1744.78	1823.71	1690.91	1983.92
tai75b	1461.37	1587.72	1464.56	1527.77	1447.39	1484.73	1480.70	1560.71	1441.35	1546.18	1509.56	1647.78
tai75c	1406.27	1527.80	1440.54	1501.91	1481.35	1664.40	1621.03	1746.07	1433.73	1502.56	1329.42	1470.60
tai75d	1430.83	1453.56	1399.83	1422.27	1414.28	1493.47	1446.50	1541.98	1408.48	1434.56	1409.14	1661.73
tai100a	2208.85	2310.37	2232.71	2295.61	2249.84	2370.58	2250.50	2462.50	2181.31	2290.95	2281.70	2550.64
tai100b	2219.28	2330.52	2147.70	2215.39	2238.42	2385.54	2169.10	2319.72	2119.03	2212.58	2255.83	2500.72
tai100c	1515.10	1604.18	1541.28	1622.66	1532.56	1627.32	1490.58	1557.81	1504.63	1589.76	1442.45	1743.07
tai100d	1881.91	2026.76	1834.60	1912.43	1955.06	2123.90	1969.94	2100.38	1793.64	1916.03	1581.36	1843.82
tai150a	3488.02	3598.69	3328.85	3501.83	3400.33	3612.79	3479.44	3680.35	3280.79	3449.32	3307.63	3684.03
tail50b	3109.23	3215.32	2933.40	3115.39	3013.99	3232.11	2934.86	3089.57	2885.94	3073.58	3128.00	3439.38
tail50c	2666.28	2913.67	2612.68	2743.55	2714.34	2875.93	2674.29	2928.77	2593.78	2759.96	2583.36	2729.15
tai150d	2950.83	3111.43	2950.61	3045.16	3025.43	3347.60	2954.64	3147.38	2911.47	3010.34	2808.99	3186.08
Average	1713.52	1807.19	1683.70	1755.10	1715.79	1844.64	1717.66	1840.95	1655.98	1742.43	1650.53	1874.29
Count		0	. 4	2	1			_	9	10	ï	•
Prop.	0.0	%00.0	10.0	%00.01	2.00%	%(5.0	2.00%	30.00%	%00	20.00%	%0

Pairs (E-ACO versus CA)	Mean difference	p value
E-ACO versus Tabu	-62.99	0.814
E-ACO versus DVRP-GA	-33.17	0.899
E-ACO versus DAPSO	-65.26	0.807
E-ACO versus VNS	-67.13	0.800

-5.45

0.983

TABLE 9: Paired *t*-test based on the published systems.

is based on fusion of ACO and K-means; meanwhile it uses 2-Opt and crossover to optimize our routes further. The algorithm is tested in a number of data sets' environments, which are derived from public VRP benchmark data. In the part of experiments, we compared the ACO, K-ACO, and E-ACO. In order to demonstrate the efficiency of proposed algorithm, the t-test is applied to perform statistical analysis. In addition, the relationship of ACO and degree of dynamism (dod) is researched by a number of tests based on different dod. With the aim of testing the performance of the approach, the experimental results are compared with previously published results. The experimental results showed that the E-ACO is able to find high-quality solutions.

Although our approach has a minor advantage in finding the best solution in DVRP, there are many issues that need to be modified. For example, the convergence rate of our approach is slower than others in later period. Therefore, various further works need to be extended. On the one hand, continuing to improve our algorithm makes it more efficient and robust. On the other hand, more complex dynamic situations, such as real-world problem application, can be solved with E-ACO.

Conflicts of Interest

E-ACO versus GA-based DVRP

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are grateful to Zhichao Ma who provided some data sets to them. This work was financially supported by Chinese National Natural Science Foundation (61572165) and Public Projects of Zhejiang Province (LGF18F030006).

References

- [1] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse, "The vehicle routing problem: State of the art classification and review," *Computers & Industrial Engineering*, vol. 99, pp. 300–313, 2016.
- [2] M. I. Hosny, Investigating Heuristic and Meta-Heuristic Algorithms for Solving Pickup and Delivery Problems, School of Computer Science Informatics, Cardiff University, Cardiff, 2010.
- [3] C. Archetti, M. G. Speranza, and D. Vigo, Vehicle Routing Problems with Profits, 2nd edition, 2014.
- [4] Y. Xu, L. Wang, and Y. Yang, "Dynamic vehicle routing using an improved variable neighborhood search algorithm," *Journal of Applied Mathematics*, vol. 2013, Article ID 672078, 2013.

- [5] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Operations Research*, vol. 21, pp. 498–516, 1973.
- [6] Hinton and T. Glyn, A thesis regarding the vehicle routing problem including a range of novel techniques for its solution, University of Bristol, 2010.
- [7] A. M. Campbell and J. H. Wilson, "Forty years of periodic vehicle routing," *Networks*, vol. 63, no. 1, pp. 2–15, 2014.
- [8] J. W. Ohlmann 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
- [9] 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.
- [10] 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, 2016.
- [11] C. Prodhon and C. Prins, "Metaheuristics for vehicle routing problems," in *Metaheuristics*, pp. 407–437, Springer, Cham, 2016.
- [12] 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.
- [13] B. Yu, 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.
- [14] C. H. Chen and C. J. Ting, "An improved ant colony system algorithm for the vehicle routing problem," *Applied Mathematics & Computation*, vol. 23, pp. 115–126, 2006.
- [15] F. T. Hanshar and B. M. Ombuki-Berman, "Dynamic vehicle routing using genetic algorithms," *Applied Intelligence*, vol. 27, no. 1, pp. 89–99, 2007.
- [16] 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, China, October 2013.
- [17] 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.
- [18] M. Okulewicz and J. Mańdziuk, "Application of particle swarm optimization algorithm to dynamic vehicle routing problem," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface, vol. 7895, no. 2, pp. 547–558, 2013.
- [19] B. Chandra Mohan and R. Baskaran, "Review: A survey: Ant Colony Optimization based recent research and implementation on several engineering domain," *Expert Systems with Applications*, vol. 39, pp. 4618–4627, 2012.
- [20] M. Dorigo and C. Blum, "Ant colony optimization theory: a survey," *Theoretical Computer Science*, vol. 344, no. 2-3, pp. 243– 278, 2005.
- [21] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behaviour," *Nature*, vol. 406, no. 6791, pp. 39–42, 2000.
- [22] S. Yang and R. Tinós, "A hybrid immigrants scheme for genetic algorithms in dynamic environments," *International Journal of Automation and Computing*, vol. 4, no. 3, pp. 243–254, 2007.

- [23] X. Yu, K. Tang, and X. Yao, "An immigrants scheme based on environmental information for genetic algorithms in changing environments," in *Proceedings of the 2008 IEEE Congress on Evolutionary Computation*, CEC 2008, pp. 1141–1147, China, June 2008
- [24] C. Li and S. Yang, "A general framework of multipopulation methods with clustering in undetectable dynamic environments," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 4, pp. 556–577, 2012.
- [25] S. Gao, Y. Wang, J. Cheng, Y. Inazumi, and Z. Tang, "Ant colony optimization with clustering for solving the dynamic location routing problem," *Applied Mathematics and Computation*, vol. 285, pp. 149–173, 2016.
- [26] S. Kashef and H. Nezamabadi-pour, "An advanced ACO algorithm for feature subset selection," *Neurocomputing*, vol. 147, no. 1, pp. 271–279, 2015.
- [27] Z. Ren, Z. Feng, L. Ke, and H. Chang, "A fast and efficient ant colony optimization approach for the set covering problem," in *Proceedings of the 2008 IEEE Congress on Evolutionary Computation, CEC 2008*, pp. 1839–1844, China, June 2008.
- [28] A. Nayyar and R. Singh, "Ant Colony Optimization (ACO) based Routing Protocols for Wireless Sensor Networks (WSN): A Survey," *International Journal of Advanced Computer Science & Applications*, 2017.
- [29] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia, "A review of dynamic vehicle routing problems," *European Journal of Operational Research*, vol. 225, no. 1, pp. 1–11, 2013.
- [30] A. Goel and V. Gruhn, "A general vehicle routing problem," European Journal of Operational Research, vol. 191, no. 3, pp. 650–660, 2008.
- [31] A. Attanasio, J.-F. Cordeau, G. Ghiani, and G. Laporte, "Parallel Tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem," *Parallel Computing*, vol. 30, no. 3, pp. 377–387, 2004.
- [32] M. Mes, M. van der Heijden, and A. van Harten, "Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems," *European Journal of Operational Research*, vol. 181, no. 1, pp. 59–75, 2007.
- [33] A. Beaudry, G. Laporte, T. Melo, and S. Nickel, "Dynamic transportation of patients in hospitals," *OR Spectrum*, vol. 32, no. 1, pp. 77–107, 2010.
- [34] J. Barceló, H. Grzybowska, and S. Pardo, "Vehicle Routing and scheduling models, simulation and City Logistics," *Operations Research/ Computer Science Interfaces Series*, vol. 38, pp. 163–195, 2007.
- [35] R. Montemanni, L. M. Gambardella, A. E. Rizzoli, and A. V. Donati, "A new algorithm for a Dynamic Vehicle Routing Problem based on Ant Colony System," in *Proceedings of the Second International Workshop on Freight Transportation Logistics*, pp. 27–30, 2003.
- [36] M. R. Garey and D. S. Johnson, "A Guide to the Theory of NP-Completeness," in *Computer Animation Conference*, 1979.
- [37] S. Ichoua, M. Gendreau, and J.-Y. Potvin, "Planned route optimization for real-time vehicle routing," *Operations Research/Computer Science Interfaces Series*, vol. 38, pp. 1–18, 2007.
- [38] A. Larsen, The Dynamic Vehicle Routing Problem, Technical University of Denmark, 2001.
- [39] B. Yu and Z. Z. Yang, "An ant colony optimization model: the period vehicle routing problem with time windows," *Transportation Research Part E: Logistics and Transportation Review*, vol. 47, no. 2, pp. 166–181, 2011.

- [40] M. C. Su and C. H. Chou, "A Modified Version of the K-Means Algorithm with a Distance Based on Cluster Symmetry," *Pattern Analysis Machine Intelligence IEEE Transactions on*, vol. 23, pp. 674–680, 2001.
- [41] M. R. Khouadjia, B. Sarasola, E. Alba, L. Jourdan, and E.-G. Talbi, "A comparative study between dynamic adapted PSO and VNS for the vehicle routing problem with dynamic requests," *Applied Soft Computing*, vol. 12, no. 4, pp. 1426–1439, 2012.
- [42] B. Ombuki, B. J. Ross, and F. Hanshar, "Multi-objective genetic algorithms for vehicle routing problem with time windows," *Applied Intelligence*, vol. 24, no. 1, pp. 17–30, 2006.
- [43] M. A. Figliozzi, "The time dependent vehicle routing problem with time windows: benchmark problems, an efficient solution algorithm, and solution characteristics," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 3, pp. 616–636, 2012.
- [44] G. A. Croes, "A method for solving traveling-salesman problems," *Operations Research*, vol. 6, pp. 791–812, 1958.
- [45] T. Vidal, M. Battarra, A. Subramanian, and G. ErdogAn, "Hybrid metaheuristics for the Clustered Vehicle Routing Problem," *Computers & Operations Research*, vol. 58, pp. 87–99, 2015.
- [46] Z. Yang, B. Yu, and C. Cheng, "A parallel ant colony algorithm for bus network optimization," *Computer-Aided Civil and Infrastructure Engineering*, vol. 22, no. 1, pp. 44–55, 2007.

















Submit your manuscripts at www.hindawi.com























