IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 22, 2021, accepted January 26, 2021, date of publication February 1, 2021, date of current version February 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3056067

An Adaptive Variable Neighborhood Search Ant Colony Algorithm for Vehicle Routing Problem With Soft Time Windows

MEILING HE [1], ZHIXIU WEI [1], XIAOHUI WU [1], AND YONGTAO PENG [2]
[1]School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China
[2]School of Management, Jiangsu University, Zhenjiang 212013, China

Corresponding author: Meiling He (hemeiling@ujs.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 71802099, and in part by the Research Projects of Philosophy and Social Sciences in Universities of Jiangsu Province under Grant 2020SJZDA062 and Grant 2020SJA2058.

ABSTRACT In this paper, an adaptive variable neighborhood search ant colony algorithm (AVNSACA) is proposed to solve the vehicle routing problem with soft time windows (VRPSTW). The ant colony algorithm's pheromone update strategy is improved to make up for the lack of pheromone in the algorithm's early stage. In order to avoid the algorithm falling into local optimum, two variable neighborhood search operators are designed, and the conditions for the algorithm to enter the variable neighborhood search are set. The effectiveness of AVNSACA in solving vehicle routing problem with soft time windows is verified by Solomon benchmark problem. Through the comparative analysis of the experimental results of the two algorithms, the advantages of the improved ant colony algorithm are illustrated. The experimental results show that the proposed algorithm can effectively obtain better solutions.

INDEX TERMS Ant colony algorithm, variable neighborhood search, adaptive, soft time window, vehicle routing problem.

I. INTRODUCTION

Vehicle routing problem with time windows (VRPTW) was first proposed by Solomon [1] in 1987. VRPTW is in line with the actual distribution situation and it is one of the most intensive combinatorial optimization problems studied in the past thirty years. Time window constraints can be divided into hard time windows and soft time windows. The difference between the two kinds of time window constraints is whether to strictly meet the time window constraints. Vehicle routing problem with hard time windows (VRPHTW) [2]–[4] requires strict compliance with time window constraints. Vehicle routing problem with soft time windows (VRPSTW) allows vehicles to arrive in advance or delay appropriately, but it needs to pay corresponding fines [5].

VRPHTW is more suitable for the actual situation of punctual arrival. Considering the high cost of providing just in time service, researchers prefer to study VRPSTW. Chiang and Russell [6] pointed out that VRPSTW can effectively reduce the delivery cost without significantly reducing customer satisfaction and can provide a feasible alternative when the hard time window is not feasible. On the basis of hard time window $[a_i, b_i]$, Taillard *et al.* [7] proposed a penalty method of soft time window based on the hard time window. When the vehicle arrives before a_i, it needs to wait until the time a_i to provide service and there is no need to pay the penalty fee. When the vehicle arrives after b_i, it needs to pay the corresponding penalty fee. Koskosidis *et al.* [8] proposed a penalty method of soft time windows, which means that no matter whether the vehicle arrives early or late, it has to pay the corresponding penalty fee. The VRPSTW studied by Chiang and Cheng [9] relaxed the time appropriately based on hard time windows constraint, that is, adding the customer's acceptable time windows. As shown in Figure 1, it shows the acceptable time window of customer i and the expected time window of customer i. If vehicle k does not arrive at customer node i within the acceptable time window, customer i rejects the service of vehicle k; if vehicle k arrives at customer i within the expected time window, customer i accepts the service of vehicle k. When vehicle $k$arrives at customer node i in $[ET_i, et_i)$ and $(lt_i, LT_i]$, it needs to pay a penalty cost, which is positively proportional to the waiting time. The penalty cost function is shown in

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Omer Farooq.

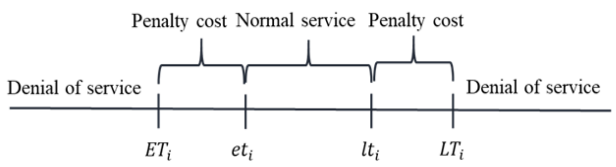

FIGURE 1. **Upper and lower limits of time window allowed to be violated.**

formula (1), where C_e is the unit penalty cost of waiting time and C_l is the unit penalty cost of late time. Based on the penalty function proposed by Chiang and Cheng [9], this paper studies the vehicle routing problem with soft time windows.

$$P(i_k) = \begin{cases} \infty & st_i^k < ET_i \\ C_e(et_i - st_i^k) & ET_i < st_i^k < et_i \\ 0 & et_i \leq st_i^k \leq lt_i \\ C_l(st_i^k - lt_i) & lt_i < st_i^k < LT_i \\ \infty & st_i^k > LT_i \end{cases} \tag{1}$$

Reducing the cost as much as possible under the premise of meeting the customer's time requirements has always been the focus of transportation enterprises and scholars. Wu *et al.* [10] established a double objective mathematical model of VRPSTW by minimizing the total transportation cost and vehicle number. Taş *et al.* [11] considered the economic point of view of the transportation company and the customers' concerns about whether the goods could be received in time, took the transportation cost and customer service cost as the optimization objectives of VRPSTW. In this paper, considering the economic benefits, the mathematical model of VRPSTW is established to minimize the vehicle distribution cost.

VRPTW belongs to the NP-hard problem. It is possible for exact algorithms to find the optimal solution within an acceptable time when the problem's scale is small, but it is difficult to solve a large scale problem in a short time. Therefore, researchers often use a metaheuristic algorithm or an improved heuristic algorithm to solve the optimal combinatorial optimization problem [12]. The metaheuristic algorithms commonly used by scholars in this field include tabu search (TS) algorithms [13], [14], simulated annealing algorithms (SAA) [15], genetic algorithms (GA) [16], ant colony algorithms (ACA) [17], and particle swarm optimization (PSO) algorithms [18]. Due to the positive feedback performance and simple operation of ACA, many scholars use this algorithm to solve the vehicle routing problem [19]–[22]. ACA was first proposed by Dorigo *et al.* [23] and applied to solve the traveling salesman problem (TSP). Chen and Ting [24] proposed an improved ant colony system algorithm (IACS), which uses new state transition rules, pheromone update rules and two local search methods, and the efficiency of IACS is verified by experiments. Yu and Yang [25] improved the performance of ACA by introducing two cross operations, and Reed *et al.* [26] used the ant colony system (ACS) to solve the capacity-constrained vehicle routing problem. Deng *et al.* [27] proposed various types of ant colony algorithm to solve VRPTW. The combination of ant colony system dealing with vehicles minimization (ACS-VEI) and max-min ant system dealing with total distance or total time minimization (MMAS-TIME) can balance good reconnaissance performance and fast convergence speed. Since traditional ACA is easy to fall into the local optimal solution [10], this paper improves the ant colony algorithm, embeds the variable neighborhood search algorithm, and proposes an adaptive variable neighborhood search ant colony algorithm (AVNSACA) to solve VRPSTW.

The other parts of this paper are arranged as follows: Section II establishes the mathematical model of VRPSTW. Section III proposes an adaptive variable neighborhood search ant colony algorithm to solve this kind of mathematical model. In section IV, an example experiment and result analysis are carried out to verify the effectiveness of the improved algorithm. Section V summarizes the research content of this paper.

II. MATHEMATICAL MODEL OF VRPSTW

The delivery vehicle routing problem with soft time windows can be described as: there are a certain number of customers, in which each customer has a cargo demand that does not exceed the maximum load of the distribution vehicle, and the distribution center has a fleet to provide distribution services for these customer points. According to the customer demand, the distribution center plans the vehicle distribution route, arranges the vehicle under the specified time window constraint condition, satisfies the customer demand, and achieves a specific goal.

In this paper, considering the economic benefits, the model's optimization goal is the vehicle distribution cost. $G = (N, E)$ is the distribution network, $N = N_D \cup N_C$ is the set of nodes, where $N_D = \{0\}$ is the distribution center, $N_C = \{1, 2, 3, \ldots, n\}$ is the set customer nodes, $E = \{(i, j)|i, j \in N, i \neq j\}$ is the edge set, which represents the directed line segment from the point i to point j, and d_{ij} is the distance between node i and node j. $K = \{1, 2, 3, \ldots, m\}$ is the vehicle set that can be deployed by the distribution center, q_i is the demand of the customer point, st_i^k is the time for vehicle k to provide service for customer i, t_{ij}^k is the travel time of vehicle k from point i to point j, and v is the average vehicle speed.

The delivery vehicle routing problem with soft time windows is abstracted as a mathematical problem based on the following assumptions:

(1) The vehicle service type is delivery.
(2) All vehicles with distribution tasks start from the distribution center and return to the distribution center at last.
(3) The demand and time window requirements of all customer points are given in advance.
(4) The coordinates of the distribution center and customers to be served are known.

(5) Each vehicle can serve multiple customer points, but only one line. The total demand of customers on the line shall not exceed its maximum passenger capacity.
(6) Each customer point can only be served by one vehicle;
(7) All vehicles are of the same type.
(8) Vehicles should serve customers within the specified time. If the arrival time of the vehicle is not within the expected time window, but within the acceptable time window, it will cause the penalty cost.

Define Variables:

$$x_{ij}^k = \begin{cases} 1, & \textit{Vehicle k travels from node i to node j} \\ 0, & \textit{else} \end{cases}$$

$$y_i^k = \begin{cases} 1, & \textit{Vehicle k serves customer i} \\ 0, & \textit{else} \end{cases}$$

Objective Function:

$$\min z = c_v \sum_{k=1}^{m}\sum_{i=0}^{n} x_{0i}^k + c_d \sum_{k=1}^{m}\sum_{i=0}^{n}\sum_{j=0}^{n} d_{ij}x_{ij}^k + \sum_{k=1}^{m}\sum_{i=0}^{n} P(i_k) \quad (2)$$

Constraints:

$$\sum_{k=1}^{m} y_i^k = 1, \quad i \in N_C \quad (3)$$

$$\sum_{j=1}^{n} x_{0j}^k = 1, \quad k \in K \quad (4)$$

$$\sum_{i=1}^{n} x_{i0}^k = 1, \quad k \in K \quad (5)$$

$$\sum_{k=1}^{m}\sum_{i=0}^{n}\sum_{j=0}^{n} x_{ij}^k \le m \quad (6)$$

$$\sum_{i=1}^{n} q_i y_i^k \le Q_v, \quad k = 1, 2 \ldots, m \quad (7)$$

$$\sum_{i=1}^{n} x_{ir}^k = \sum_{j=1}^{n} x_{rj}^k, \quad r \in N_C,\ k \in K \quad (8)$$

$$t_i + st_i^k + t_i^k \le st_j^k + M\left(1 - x_{ij}^k\right), \quad i, j \in E \quad (9)$$

$$x_{ij}^k \in \{0, 1\}, \quad (i, j) \in E,\ k \in K \quad (10)$$

$$y_i^k \in \{0, 1\}, \quad i \in Nk \in K \quad (11)$$

Equation (2) is the objective function to minimize the vehicle distribution cost. Constraint (3) indicates that each customer is served by at most one vehicle. Constraint (4) indicates that each vehicle starts from the distribution center. Constraint (5) indicates that all vehicles must return to the distribution center at last. Constraint (6) indicates that the number of vehicles dispatched shall not exceed the maximum number of vehicles available for the fleet. Constraint (7) means that the demand of all customers served by each vehicle shall not exceed the maximum capacity limit. Constraints (8) ensure the same number of vehicles in and out of any customer point. Constraint (9) represents the time relationship of vehicle arrival between two adjacent nodes on a route, M is a large constant. Constraints (10) and (11) show the value range of each variable.

III. THE PROPOSED ALGORITHM

Ant colony algorithm (ACA) is an optimization algorithm for simulating ant foraging behavior. The basic principle of ACA is that ants release pheromones on the path they pass. If ants encounter a road junction that they have not passed, they choose a path randomly and release pheromones. The pheromone concentration released is inversely proportional to the length of the path. Ants in the ant colony can perceive pheromone and walk along the path with high pheromone concentration, forming a positive feedback mechanism. After some time, all ants will walk along the shortest path to find food sources. The two main steps of the ant colony algorithm are to construct ant routes and update pheromone [26].

Construct ant routes. Ant k starts from the distribution center and selects the next node by roulette. The probability of ant k moving from node i to node j is as follows:

$$p_{ij}^k = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t) \cdot \eta_{ij}^{\beta}(t)}{\sum_{r \in A_k} \tau_{ir}^{\alpha}(t) \cdot \eta_{ir}^{\beta}(t)}, & j \in allowed_k \\ 0, & else \end{cases} \quad (12)$$

In equation (12), α is the pheromone importance factor, β is the heuristic importance factor, τ_{ij} represents the pheromone released by ants on edge (i, j), and η_{ij} is the visibility of the edge (i, j), η_{ij}=1/d_{ij}.

Update pheromone. The pheromone updating of the path is required after each iteration of the ant colony algorithm, and the update rules are as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (13)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^k(t) \quad (14)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \dfrac{Q}{L_k}, & \textit{ant k passes through } (i, j) \\ 0, & \textit{else} \end{cases} \quad (15)$$

The $\Delta\tau_{ij}(t)$ in equation (14) is the pheromone increment on edge (i, j) in the t-th iteration, and equation (15) is the Ant-Cycle Model (ACM), which represents the pheromone content marked by ant k on edge (i, j) in this iteration.

A. IMPROVEMENT OF PHEROMONE UPDATING STRATEGY

The ant colony algorithm is easy to fall into local optimization. Pheromone updating strategy is one of the critical steps of the ant colony algorithm. If the pheromone volatilization coefficient is too small, the randomness of algorithm search will be reduced, leading to the algorithm falling into the local optimal solution or even stagnation. If the pheromone volatilization coefficient is too large, the convergence speed of the algorithm will be slow. Therefore, by adjusting the volatility coefficient of pheromone, the algorithm has robust searchability in the early stage and can accelerate the later stage's convergence speed. The trigger condition of adjusting the pheromone volatilization coefficient is that the subtraction value of the target value obtained by two successive iterations is less than 0.001. If the trigger condition is satisfied, the pheromone volatilization coefficient is adjusted according to equation (16).

$$\rho = \begin{cases} max\{0.95 \cdot \rho, \rho_{min}\} & iter > 1 \\ \rho_0 & iter = 1 \end{cases} \tag{16}$$

Among them, ρ_{min} is the lower limit of pheromone volatilization coefficient, and ρ_0 is the initial pheromone volatilization coefficient.

In this paper, the Ant-Cycle Model in ACA is improved to reduce the pheromone distribution on the edge with high total transportation cost and increase the pheromone increment on the edge with a lower total cost. The expression is as follows (where Q is the constant of updating pheromone concentration, z is the target value of the kth ant in this iteration, and z_{max} is the highest target value obtained in this iteration):

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \dfrac{z_{max} - z}{z} \cdot Q, & ant\ k\ passes\ through\ (i, j) \\ 0, & else \end{cases} \tag{17}$$

B. INSERTION OF VARIABLE NEIGHBORHOOD LOCAL SEARCH ALGORITHM

Variable neighborhood search algorithm (VNS) is a general local search algorithm proposed by Mladenović and Hansen [28]. Based on the design of local search operator, we can use the characteristics of the existing optimal solution to get a better neighborhood solution, and reuse the local search operator to get the local optimal solution. The general variable neighborhood algorithm is divided into two parts: shaking and variable neighborhood descent search (VND). The initial solution is generated in the shaking stage, and the initial solution is further optimized by using different neighborhood structure operators in the variable neighborhood descent search stage. In this paper, the ant colony algorithm generated the initial solution, so the shaking step is omitted. The specific sinking search step is shown in Figure 2. Define m neighborhood $N_i(S)$ and search in the neighborhood structure i. If a better solution S' than the initial solution S is found, then $S = S'$, the algorithm will continue to search in this neighborhood until no better solution is found. The next neighborhood structure will be transferred, and the above operation will be repeated. If all m neighborhood structures are searched and no better solution is found, the neighborhood search is finished and the optimal solution is output.

In this paper, two neighborhood structures are used for local search. Swap operator and insertion operator are designed respectively, as shown in Figure 3 and Figure 4.

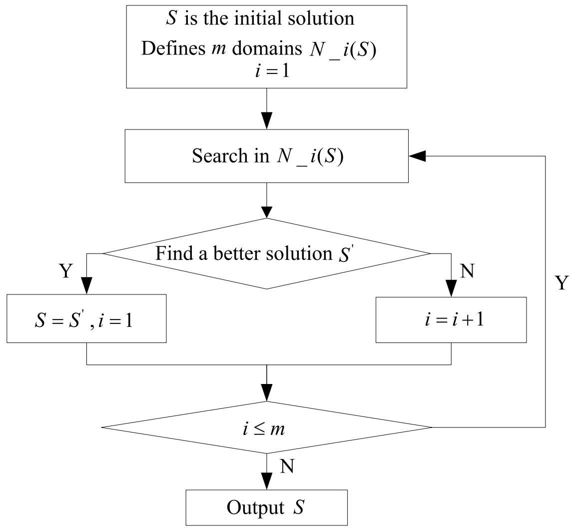

FIGURE 2. Flow chart of VND.

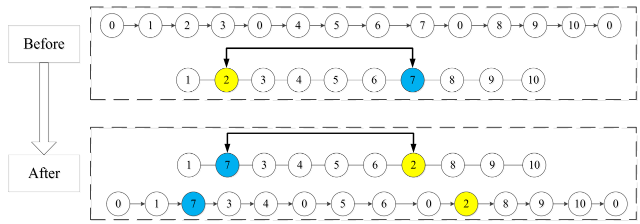

FIGURE 3. Swap operator in neighborhood structure 1.

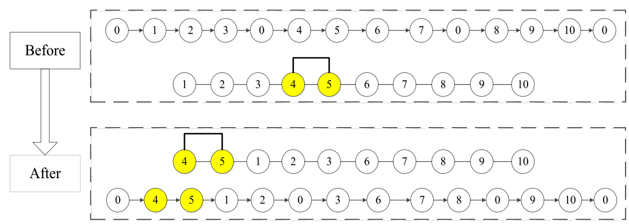

FIGURE 4. Insertion operator in neighborhood structure 2.

Node 0 represents the distribution center, and other numbers represent customer nodes. The initial solution is three paths: 0-1-2-3-0, 0-4-5-6-7-0, and 0-8-9-10-0.

1) SWAP OPERATOR

As shown in Figure 3, the swap operator is to select two customer nodes to exchange positions randomly in all routes of the current solution S. Firstly, delete the distribution center node "0" from all routes, and retain the original order of the remaining customer nodes (1-2-3-4-5-6-7-8-9-10). Then select two nodes to swap randomly (node 2 and node 7 are selected in the figure, and the order after the swap is 1-7-3-4-5-6-2-8-9-10). Finally, according to the capacity and time constraints, the distribution center nodes "0" is reinserted to get three new routes (0-1-7-3-4-0, 0-5-6-0, 0-2-8-9-10-0). If the new solution S' is better than S, then $S = S'$, and repeat the above operation until no better solution is found or all the

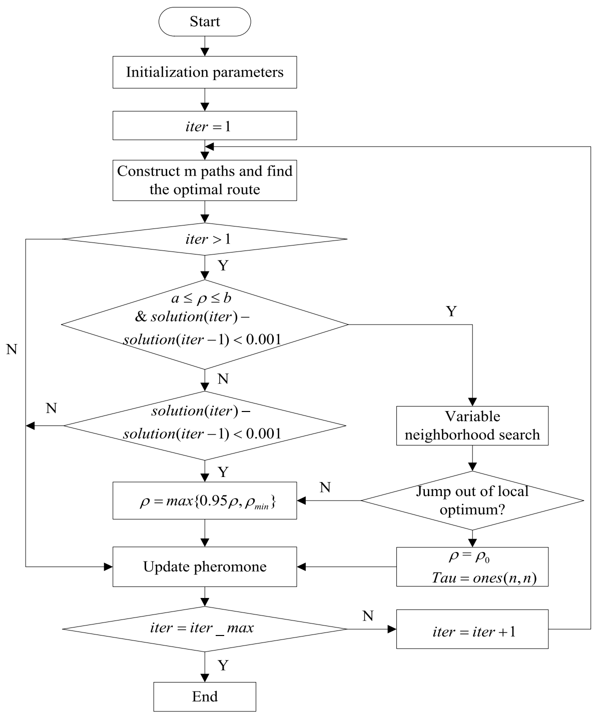

FIGURE 5. Flow chart of AVNSACA.

swap combination attempts are completed, then the insertion operator is used for the solution S.

2) INSERTION OPERATOR

As shown in Figure 4, the insertion operator is to select two adjacent customer nodes randomly from all routes of the current solution S and insert them into the front end of the remaining customer nodes in order. Firstly, delete the distribution center node "0" from all routes, and keep the original order of the remaining customer nodes (1-2-3-4-5-6-7-8-9-10). Then randomly select two adjacent nodes and insert them into the front end of the remaining customer points in order (adjacent nodes 4 and 5 are selected in the figure, and the inserted order is 4-5-1-2-3-6-7-8-9-10). Finally, three new routes (0-4-5-1-2-0, 0-3-6-7-8-9-0, 0-9-10-0) are generated by reinserting the distribution center node "0" according to the capacity and time constraints. If the new solution S' is better than S, then $S = S'$, the insertion operator is used for the solution S. If no better solution is found or all the node combination attempts are completed, exit the local search.

C. ADAPTIVE VARIABLE NEIGHBORHOOD SEARCH ANT COLONY ALGORITHM

The steps of the adaptive variable neighborhood search ant colony algorithm are shown in Figure 5. When $\rho > b$, the algorithm does not perform local search. When $a \leq \rho \leq b$, the variable neighborhood local search algorithm is inserted to make it jump out of the local optimum. When the local search is used to jump out of the local optimum, let $\rho = \rho_0$, reset the pheromone matrix, expand the search scope and randomness of the algorithm again. When $\rho < a$, the local search is exited to reduce the running time of the algorithm.

IV. EXAMPLE ANALYSIS

All the simulation experiments in this paper are written with MATLAB R2018b and run under Intel Core i5-3210m 2.5GHz (4.00GB RAM), windows 10 operating system.

A. EXAMPLE SELECTION AND PARAMETER SETTING

Solomon benchmark is the data that reflects the real-life scheduling situation proposed by Solomon in 1987. It is the most commonly used standard test set to evaluate the performance of the VRPTW algorithm. There are three kinds of customer sizes, namely 25, 50 and 100 customers. Each scale has 56 sets of test data, including six types: C1, C2, R1, R2, RC1, and RC2. The locations of customers in type-C instances are geographically clustered, while those in type-R instances are randomly distributed. The locations of customers in type-RC instances have both clustering and random distribution characteristics. The customer time windows of type-C1, type-R1, and type-RC1 instances are narrow, and the vehicle load is small. Type-C2, type-R2 and type-RC2 have more relaxed customer time windows and larger vehicle loads compared with type-C1, type-R1, and type-RC1. Under the condition that the time window is allowed to be relaxed, the impact of the results is more evident in the case of a tight customer time window. This paper takes 100 customer nodes as the scale [29], and selects 4 instances of type-C1, type-R1, and type-RC1, a total of 12 instances. The experimental analysis is carried out under the condition that the time window is allowed to be relaxed.

The parameters are set as follows: ant number $m = 80$, maximum iteration times $iter\_max = 60$, pheromone concentration increment coefficient $Q = 10$, pheromone importance factor $\alpha = 1$, heuristic importance factor $\beta = 2$, pheromone volatilization coefficient $\rho = 0.8$, $\rho_{min} = 0.01$, $a = 0.4$, $b = 0.5$.

B. RESULT ANALYSIS

The adaptive variable neighborhood search ant colony algorithm is used to test the 12 instances and each instance is tested 10 times. Table 1 records the best-know solution with hard time windows constraint, and the optimal solution obtained with soft time windows constraint by this algorithm. Where TD is the total distance, NV is the number of vehicles, $Time_{sec}$ is the running time (the unit is seconds), Gap_{TD} represents the gap the distance of the optimal solution obtained by this algorithm and the distance of the known optimal solution, $Gap_{TD} = (TD_2 - TD_1)/TD_1 \times 100\%$. Gap_{NV} is the gap between the number of vehicles in the optimal solution obtained by this algorithm and the

TABLE 1. Comparison between the optimal solution obtained by AVNSACA and the best-known solution.

Dataset	Best-known[29]		AVNSACA			$Gap_{TD}(\%)$	$Gap_{NV}(\%)$
	TD_1	NV_1	TD_2	NV_2	$Time_{sec}$		
C101	828.94	10	828.94	10	34.32	0.00	0.00
C105	828.94	10	828.94	10	29.03	0.00	0.00
C106	828.94	10	828.94	10	31.70	0.00	0.00
C109	828.94	10	828.02	10	37.64	0.11	0.00
R101	1650.80	19	1384.83	15	44.51	-16.11	-21.05
R102	1486.12	17	1367.79	14	49.16	-7.96	-17.65
R103	1292.68	13	1263.70	13	48.60	-2.24	0.00
R105	1377.11	14	1330.47	14	51.54	-3.39	0.00
RC101	1696.95	14	1483.62	13	44.84	-12.57	-7.14
RC102	1554.75	12	1436.57	13	49.82	-7.60	8.33
RC105	1629.44	13	1405.60	13	48.16	-13.74	0.00
RC106	1424.73	11	1333.48	12	46.32	-6.41	9.09

TABLE 2. Results of solving 12 instances by two algorithms.

Number of calculations	ACA				AVNSACA				$Gap_{best}(\%)$	$Gap_{ave}(\%)$
	$Cost_1$	$Tw_1(\%)$	$Cost_{1ave}$	$Cost_{1SD}$	$Cost_2$	$Tw_2(\%)$	$Cost_{2ave}$	$Cost_{2SD}$		
C101	983.76	98	1130.16	108.59	928.94	100	957.02	23.89	-5.57	-15.32
C105	1063.04	96	1197.68	79.13	928.94	100	967.45	25.34	-12.61	-19.22
C106	1212.18	96	1366.38	97.91	1052.12	99	1085.23	23.43	-13.20	-20.58
C109	1283.52	97	1385.37	58.39	984.28	100	1044.37	29.80	-23.31	-24.61
R101	2443.33	35	2480.72	22.14	1983.47	49	2021.61	21.13	-18.82	-18.51
R102	2301.97	44	2364.68	30.58	1837.56	65	1875.43	21.98	-20.17	-20.69
R103	1998.85	64	2046.59	29.54	1643.38	79	1679.43	21.13	-17.78	-17.94
R105	2077.82	50	2162.48	44.95	1665.19	82	1710.48	21.03	-19.86	-20.90
RC101	2458.78	52	2517.49	42.57	1918.69	67	1951.60	16.62	-21.97	-22.48
RC102	2217.42	71	2392.08	80.44	1747.95	82	1777.84	26.33	-21.17	-25.68
RC105	2231.17	65	2373.16	57.46	1800.10	81	1876.25	33.89	-19.32	-20.94
RC106	2033.49	68	2126.59	58.59	1654.90	94	1702.86	33.01	-18.62	-19.93

number of vehicles with the best-known solution, $Gap_{NV} = (NV_2 - NV_1)/NV_1 \times 100\%$.

According to the results in Table 1, there are three groups of distances of type-C1 instance, which are the same as the best-known solutions. The optimal distance of C109 is slightly better than the best-known solutions. Although the customer time window is narrow in type-C1 instance, it is relatively loose compared with type-R1 and type-RC1 instance. Therefore, when the time window is relaxed, the effect of the result is not apparent. In the eight instances of type-R1 and type-RC1, the results of AVNSACA are better than the best-known solutions in the optimal distance. Compared with the results of the best-known solutions, the distance optimization rates of R101, RC101, RC105 are more than 10%. For the optimal number of vehicles, the results of R101, R102, and RC101 obtained by this algorithm are better than the best-known solutions. Except for RC102 and RC106, the optimal number of vehicles in other instances is equal to or better than the best-known solutions. Although the number of vehicles of RC102 and RC106 calculated by this algorithm is one more than the best-known solutions, the vehicle distance has apparent advantages. Compared with the results in Table 1, it can be seen that this algorithm can effectively solve the vehicle routing problem with soft time windows and has obvious advantages.

TABLE 3. Optimal solution and specific distribution scheme of C105.

		ACA	AVNSACA
Optimal cost		1063.04	928.94
Optimal distribution route	1	0-13-17-18-19-15-16-14-12-99-0	0-98-96-95-94-92-93-97-100-99-0
	2	0-5-3-7-8-10-11-9-6-4-2-1-75-0	0-57-55-54-53-56-58-60-59-0
	3	0-20-24-25-27-29-30-28-26-23-22-21-49-47-0	0-81-78-76-71-70-73-77-79-80-0
	4	0-67-65-63-62-74-72-61-64-68-66-69-0	0-43-42-41-40-44-46-45-48-51-50-52-49-47-0
	5	0-43-42-41-40-44-46-45-48-51-50-52-0	0-20-24-25-27-29-30-28-26-23-22-21-0
	6	0-55-57-54-53-56-58-60-59-0	0-13-17-18-19-15-16-14-12-0
	7	0-32-33-31-35-37-38-39-36-34-0	0-5-3-7-8-10-11-9-6-4-2-1-75-0
	8	0-81-78-76-71-70-73-77-79-80-0	0-67-65-63-62-74-72-61-64-68-66-69-0
	9	0-90-87-86-83-82-84-85-88-89-91-0	0-90-87-86-83-82-84-85-88-89-91-0
	10	0-98-96-95-94-93-92-97-100	0-32-33-31-35-37-38-39-36-34-0

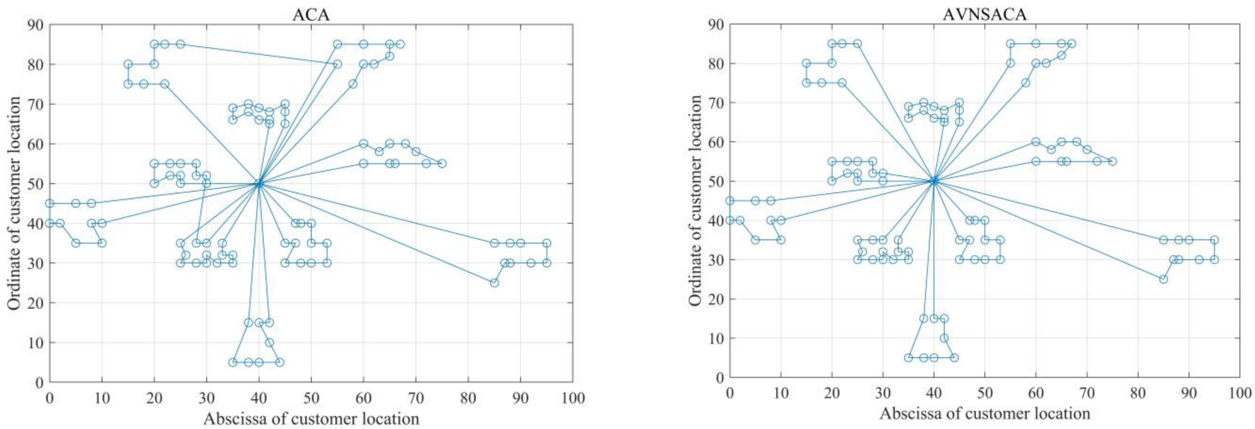

FIGURE 6. Schematic diagram of the optimal distribution path of two algorithms.

C. COMPARATIVE ANALYSIS OF ALGORITHMS

ACA and AVNSACA are used to solve the instances in Table 1, and the results are compared and analyzed. For the convenience of calculation, the vehicle cost coefficient is taken as 10, and other cost coefficients are taken as 1. In practical problems, the unit cost should be determined according to the specific situation. The parameters of the two algorithms are consistent with that in part A of section IV. The two algorithms run 10 times for each instance respectively. The running results of the algorithm are shown in Table 2. Among them, $Cost$ is the minimum distribution cost, $Cost_{ave}$ is the average distribution cost, $Cost_{SD}$ is the standard deviation of distribution cost, Tw is the punctuality rate of vehicles arriving at the customer point under the minimum cost scenario, and Gap_{best} is the gap between the minimum vehicle distribution costs calculated by ACA and AVNSACA, $Gap_{best} = (Cost_2 - Cost_1)/Cost_1 \times 100\%$, Gap_{ave} is the gap between the average vehicle distribution costs calculated by ACA and AVNSACA, $Gap_{ave} = (Cost_{2ave} - Cost_{1ave})/Cost_{1ave} \times 100\%$.

The optimization schemes of the two algorithms are compared. The cost of instances obtained by AVNSACA is obviously less than that of ACA, and the cost optimization rate of AVNSACA's 11 instances is more than 10%. For type-R1 and type-RC1 instances, the optimization effect of AVNSACA is obvious, and the cost optimization rate is above 15%. The punctuality rate of AVNSACA's vehicles arriving at the customer's point is also higher than that of ACA, which shows that the algorithm proposed in this paper can give a distribution scheme with higher customer satisfaction and lower cost than ACA.

In terms of the average cost of the two algorithms, all the results obtained by AVNSACA is lower than that of ACA, and the difference is more than 15%. In terms of standard deviation of the two algorithms, it can be found that there are great differences between type-C1 and type-RC1 instances, which shows that the robustness of AVNSACA is better than that of ACA. For type-R1 instances with a random distribution of customer points, the solution space of the algorithm is more complex, so the cost standard deviations obtained by the two algorithms are similar. Overall, the optimization effect and robustness of AVNSACA are better than ACA.

Compared with ACA, the running time of AVNSACA is increased. Since the improved algorithm embeds the variable neighborhood search algorithm in the ant colony algorithm, it can improve the solution accuracy as well as increase the

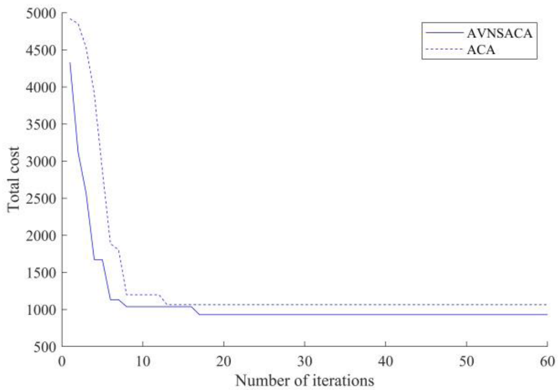

FIGURE 7. Comparison of convergence curves of two algorithms.

computational complexity. However, the running time of both algorithms is less than 60s, from the optimization effect of the target value obtained by the improved algorithm, we think that the increased running time is in a reasonable range.

C105 is taken as an example to show the specific optimal distribution scheme of the two algorithms. Table 3 shows the optimal solution and the specific distribution scheme of C105. Figure 6 shows the optimal distribution route of ACA and AVNSACA, which indicates that the latter is better. Figure 7 displays the comparison of the two algorithms' convergence curves to obtain the optimal solution. It can be seen that AVNSACA has a faster optimization speed in the early stage. In the iteration interval (10,20), the two algorithms fall into local optimum, but AVNSACA jumps out of local optimum by inserting variable neighborhood local search operation, while ACA does not. Therefore, the adaptive variable neighborhood search ant colony algorithm proposed in this paper is effective.

V. CONCLUSION

Aiming at the shortcomings of traditional ant colony algorithms, such as lack of pheromone, slow search speed, and easy to fall into local optimum, an adaptive variable neighborhood search ant colony algorithm is proposed to solve the problem model in this paper. The pheromone updating strategy is improved, and the variable neighborhood local search algorithm is embedded to improve the solution accuracy. The validity of the algorithm in solving VRPSTW is verified by 12 instances in Solomon benchmark. Under the same conditions, AVNSACA and ACA are used to calculate the above 12 instances respectively. The results show that AVNSACA can achieve better results and stronger robustness than ACA. To sum up, the adaptive variable neighborhood search ant colony algorithm proposed in this paper can effectively solve the distribution vehicle routing problem with soft time window constraints. Compared with the traditional ant colony algorithm, it can obtain a path planning scheme with lower cost and higher service quality.

REFERENCES

[1] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Oper. Res.*, vol. 35, no. 2, pp. 254–265, Apr. 1987, doi: 10.1287/opre.35.2.254.

[2] G. Andreatta, M. Casula, C. D. Francesco, and L. D. Giovanni, "A branch-and-price based heuristic for the stochastic vehicle routing problem with hard time windows," *Electron. Notes Discrete Math.*, vol. 52, pp. 325–332, Jun. 2016, doi: 10.1016/j.endm.2016.03.043.

[3] R. Pérez-Rodríguez and A. Hernández-Aguirre, "A hybrid estimation of distribution algorithm for the vehicle routing problem with time windows," *Comput. Ind. Eng.*, vol. 130, pp. 75–96, Apr. 2019, doi: 10.1016/j.cie.2019.02.017.

[4] C. Hu, J. Lu, X. Liu, and G. Zhang, "Robust vehicle routing problem with hard time windows under demand and travel time uncertainty," *Comput. Oper. Res.*, vol. 94, pp. 139–153, Jun. 2018, doi: 10.1016/j.cor.2018.02.006.

[5] M. A. Figliozzi, "An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows," *Transp. Res. C, Emerg. Technol.*, vol. 18, no. 5, pp. 668–679, Oct. 2010, doi: 10.1016/j.trc.2009.08.005.

[6] W.-C. Chiang and R. A. Russell, "A Metaheuristic for the vehicle-routeing problem with soft time windows," *J. Oper. Res. Soc.*, vol. 55, no. 12, pp. 1298–1310, Dec. 2004, doi: 10.1057/palgrave.jors.2601791.

[7] É. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J.-Y. Potvin, "A tabu search heuristic for the vehicle routing problem with soft time windows," *Transp. Sci.*, vol. 31, no. 2, pp. 170–186, May 1997, doi: 10.1287/trsc.31.2.170.

[8] Y. A. Koskosidis, W. B. Powell, and M. M. Solomon, "An optimization-based heuristic for vehicle routing and scheduling with soft time window constraints," *Transp. Sci.*, vol. 26, no. 2, pp. 69–85, May 1992, doi: 10.1287/trsc.26.2.69.

[9] W. C. Chiang and C. Y. Cheng, "Considering the performance bonus balance in the vehicle routing problem with soft time windows," *Procedia Manuf.*, vol. 11, no. 9, pp. 2156–2163, 2017, doi: 10.1016/j.promfg.2017.07.361.

[10] L. Wu, Z. He, Y. Chen, D. Wu, and J. Cui, "Brainstorming-based ant colony optimization for vehicle routing with soft time windows," *IEEE Access*, vol. 7, pp. 19643–19652, 2019, doi: 10.1109/ACCESS.2019.2894681.

[11] D. Taş, N. Dellaert, T. van Woensel, and T. de Kok, "Vehicle routing problem with stochastic travel times including soft time windows and service costs," *Comput. Oper. Res.*, vol. 40, no. 1, pp. 214–224, Jan. 2013, doi: 10.1016/j.cor.2012.06.008.

[12] M. M. A. El Aziz, H. A. El-Ghareeb, and M. S. M. Ksasy, "Review in finding path problem classification and algorithms in vehicle routing," *Int. J. Inf. Sci. Intell. Syst.*, vol. 3, no. 2, p. 20, 2014.

[13] G. Li and J. Li, "An improved tabu search algorithm for the stochastic vehicle routing problem with soft time windows," *IEEE Access*, vol. 8, pp. 158115–158124, 2020, doi: 10.1109/ACCESS.2020.3020093.

[14] Y. Wu, B. Zheng, and X. Zhou, "A disruption recovery model for time-dependent vehicle routing problem with time windows in delivering perishable goods," *IEEE Access*, vol. 8, pp. 189614–189631, 2020, doi: 10.1109/ACCESS.2020.3032018.

[15] Y. Shi, Y. Zhou, W. Ye, and Q. Q. Zhao, "A relative robust optimization for a vehicle routing problem with time-window and synchronized visits considering greenhouse gas emissions," *J. Cleaner Prod.*, vol. 275, Dec. 2020, Art. no. 124112, doi: 10.1016/j.jclepro.2020.124112.

[16] T.-S. Khoo, B. B. Mohammad, V.-H. Wong, Y.-H. Tay, and M. Nair, "A two-phase distributed ruin- and-recreate genetic algorithm for solving the vehicle routing problem with time windows," *IEEE Access*, vol. 8, pp. 169851–169871, 2020, doi: 10.1109/ACCESS.2020.3023741.

[17] H. Mao, J. Shi, Y. Zhou, and G. Zhang, "The electric vehicle routing problem with time windows and multiple recharging options," *IEEE Access*, vol. 8, pp. 114864–114875, 2020, doi: 10.1109/ACCESS.2020.3003000.

[18] Y. Marinakis, M. Marinaki, and A. Migdalas, "A multi-adaptive particle swarm optimization for the vehicle routing problem with time windows," *Inf. Sci.*, vol. 481, pp. 311–329, May 2019, doi: 10.1016/j.ins.2018.12.086.

[19] Y. Li, H. Soleimani, and M. Zohal, "An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives," *J. Cleaner Prod.*, vol. 227, pp. 1161–1172, Aug. 2019, doi: 10.1016/j.jclepro.2019.03.185.

[20] S. R. Balseiro, I. Loiseau, and J. Ramonet, "An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows," *Comput. Oper. Res.*, vol. 38, no. 6, pp. 954–966, Jun. 2011, doi: 10.1016/j.cor.2010.10.011.

[21] Y. Wang, L. Wang, G. Chen, Z. Cai, Y. Zhou, and L. Xing, ''An improved ant colony optimization algorithm to the periodic vehicle routing problem with time window and service choice,'' *Swarm Evol. Comput.*, vol. 55, Jun. 2020, Art. no. 100675, doi: 10.1016/j.swevo.2020.100675.
[22] P. Stodola, ''Hybrid ant colony optimization algorithm applied to the multi-depot vehicle routing problem,'' *Natural Comput.*, vol. 19, no. 2, pp. 463–475, Jun. 2020, doi: 10.1007/s11047-020-09783-6.
[23] M. Dorigo, G. D. Caro, and L. M. Gambardella, ''Ant algorithms for discrete optimization,'' *Artif. Life*, vol. 5, no. 2, pp. 137–172, Apr. 1999, doi: 10.1162/106454699568728.
[24] C.-H. Chen and C.-J. Ting, ''An improved ant colony system algorithm for the vehicle routing probleM,'' *J. Chin. Inst. Ind. Eng.*, vol. 23, no. 2, pp. 115–126, Jan. 2006, doi: 10.1080/10170660609509001.
[25] B. Yu and Z. Z. Yang, ''An ant colony optimization model: The period vehicle routing problem with time windows,'' *Transp. Res. E, Logistics Transp. Rev.*, vol. 47, no. 2, pp. 166–181, Mar. 2011, doi: 10.1016/j.tre.2010.09.010.
[26] M. Reed, A. Yiannakou, and R. Evering, ''An ant colony algorithm for the multi-compartment vehicle routing problem,'' *Appl. Soft Comput.*, vol. 15, pp. 169–176, Feb. 2014, doi: 10.1016/j.asoc.2013.10.017.
[27] Y. Deng, W. Zhu, H. Li, and Y. Zheng, ''Multi-type ant system algorithm for the time dependent vehicle routing problem with time windows,'' *J. Syst. Eng. Electron.*, vol. 29, no. 3, pp. 625–638, 2018, doi: 10.21629/JSEE.2018.03.20.
[28] N. Mladenović and P. Hansen, ''Variable neighborhood search,'' *Comput. Oper. Res.*, vol. 24, no. 11, pp. 1097–1100, Nov. 1997.
[29] *Best Known Solution Values*. Accessed: Nov. 2, 2020. [Online]. Available: https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark/100-customers/

MEILING HE received the Ph.D. degree in logistics engineering from Southwest Jiaotong University, China, in 2010. She is currently an Associate Professor with the School of Automotive and Traffic Engineering, Jiangsu University. Her current research interests include transportation system optimization, logistics system planning, and reverse logistics.

ZHIXIU WEI received the B.E. degree from the School of Jingjiang, Jiangsu University, Zhenjiang, China, in 2018. She is currently pursuing the M.E. degree in transportation engineering with the School of Automotive and Traffic Engineering, Jiangsu University. Her research interest includes vehicle routing optimization.

XIAOHUI WU received the M.E. degree in transportation planning and management from Southwest Jiaotong University, China, in 2008. He is currently pursuing the Ph.D. degree in traffic engineering. He is also a Lecturer with the School of Automotive and Traffic Engineering, Jiangsu University. His current research interests include optimization of urban transportation systems and analysis of residents' travel behavior.

YONGTAO PENG received the Ph.D. degree in logistics engineering from Southwest Jiaotong University, China, in 2014. He is currently an Associate Professor with the School of Management, Jiangsu University. His current research interests include transportation system optimization, logistics and supply chain management, production operation management, and service science.