

Research Article

Learning Whale Optimization Algorithm for Open Vehicle Routing Problem with Loading Constraints

Nai K. Yu ^{1,2}, Wen Jiang ², Rong Hu ^{2,3}, Bin Qian ^{1,2,3} and Ling Wang⁴

¹Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology, Kunming 650500, China

²Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China

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

⁴Department of Automation, Tsinghua University, Beijing 100084, China

Correspondence should be addressed to Rong Hu; ronghu@vip.163.com

Received 28 July 2021; Revised 30 November 2021; Accepted 6 December 2021; Published 26 December 2021

Academic Editor: Lianbo Ma

Copyright © 2021 Nai K. Yu 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.

This paper addresses the two-dimensional loading open vehicle routing problem with time window (2L-OVRPTW). We propose a learning whale optimization algorithm (LWOA) to minimize the total distance; an improved skyline filling algorithm (ISFA) is designed to solve the two-dimensional loading problem. In LWOA, the whale optimization algorithm is used to search the solution space and get the high-quality solution. Then, by learning and accumulating the block structure and customer location information in the high-quality solution individuals, a three-dimensional matrix is designed to guide the updating of the population. Finally, according to the problem characteristics, the local search method based on fleet and vehicle is designed and performed on the high-quality solution region. IFSA is used to optimize the optimal individual. The computational results show that the proposed algorithm can effectively solve 2L-OVRPTW.

1. Introduction

Vehicle routing problem (VRP), as a classical combinatorial optimization problem, was first proposed by Dantzig and Ramser [1]. Open vehicle routing problem (OVRP) is a variant of VRP, which widely exists in the third-party logistics distribution. This type of problem is generating a set of vehicle routes that meet vehicle load conditions, compartment volume, and customer service demand. The vehicles do not need to return to the central depot. With the rapid development of e-commerce technology, the importance of third-party logistics is more prominent, and the loading link is an essential part of the logistics field. The loading order of goods can directly affect the service efficiency of customer points. In this context, it has important practical significance to study the two-dimensional loading open vehicle routing problem with time windows (2L-OVRPTW). 2L-OVRPTW includes “routing” and “loading.” Generally speaking, we need to consider both “routing” and “loading” to obtain a satisfactory solution. In

addition, due to factors such as the weight, height, and fragility of the goods, the goods cannot be stacked on each other during the loading process, such as the distribution of large household appliances and delicate home appliances. Therefore, the study of 2L-OVRPTW has an important economic value. In theory, 2L-OVRPTW comprises a vehicle routing problem and a two-dimensional loading problem, so 2L-OVRPTW is an NP-hard problem with high complexity.

The research on the vehicle routing problem and the two-dimensional loading problem has been relatively mature, but the vehicle routing problem with two-dimensional loading constraints (2L-CVRP) has gradually received attention in recent years. Research is very limited. Ma et al. [2] proposed a graph-based fuzzy evolutionary algorithm to solve the two-echelon vehicle-routing problem. With the optimization goal of minimizing the cost, each offspring is generated from the parent’s allocation graph through the graph-based fuzzy allocation process, and a fuzzy local search process is designed to further improve the performance of the offspring. Iori et al. [3] proposed 2L-CVRP for

the first time in 2007 and used an exact algorithm based on branch pricing to solve it, but this algorithm can only effectively solve small- and medium-sized cases. Zachariadis et al. [4] proposed a metaheuristic algorithm for 2L-CVRP. The algorithm combines the tabu search algorithm and the guiding local search principle. In addition, the paper uses storage structure record feasible loading scheme and a set of heuristic algorithms to solve the loading problem and has achieved good results. Fuellerer et al. [5] proposed a new ant colony optimization algorithm based on saving algorithm to solve 2L-CVRP. At the same time, different heuristic algorithms were used to solve the loading problem. Good results were obtained on small-scale issues. Duhamel et al. [6] proposed a $\text{GRID} \times \text{ELS}$ algorithm for 2L-CVRP, which transforms loading constraints into resource constrained project scheduling problems. The optimization framework deals with RCPSP-CVRP and finally transforms the RCPSP-CVRP solution into 2L-CVRP solution by solving special loading problems. For 2L-CVRP, Guimarans et al. [7] proposed a hybrid simulation algorithm combining Monte Carlo simulation, an iterative local search framework, and a route and bin loading heuristic method with stochastic bias to solve the problem. Wei et al. [8] proposed a variable domain search algorithm to solve the routing problem for 2L-CVRP, used skyline heuristic algorithm to solve the loading problem, and designed a data structure to record the loading feasibility information to reduce the loading time. Leung et al. [9] proposed a metaheuristic method combining tabu search and extended guided local search (EGLS). Firstly, it is proved that tabu search algorithm can solve CVRP effectively, and EGLS can avoid the algorithm from falling into local optimal. Secondly, a new heuristic loading algorithm is added to solve the loading problem, which improves the distribution cost significantly. Wei et al. [10], aiming at 2L-CVRP, proposed a simulated annealing algorithm with repeated temperature rise and fall mechanism to solve the problem and adopted a heuristic method based on open space to solve the loading problem. Based on 2L-CVRP, Rui et al. [11] established the corresponding mathematical model considering the constraints of multi-depot and time window. The former is used to solve the vehicle routing problem, and the latter is used to solve the bin loading problem. Two groups of experiments are designed to verify the effectiveness of the algorithm. Wang and Zhou [12] aimed at establishing a multiobjective vehicle scheduling problem model with two-dimensional loading constraints based on comprehensively considering the constraints of time window, two-dimensional loading, and customer satisfaction. At the same time, a multiobjective ant colony optimization algorithm was proposed. In the loading stage, the two-dimensional loading strategy of improved minimum horizontal line search algorithm was adopted to improve the success rate of loading. Ma et al. [2] proposed an adaptive localized decision variable analysis approach under the decomposition-based framework to solve large-scale multiobjective optimization problems. An adaptive strategy is used to optimize the decision variables, which can adaptively balance the convergence and diversity of solutions in the target space. Leung et al. [13] established a

mathematical model to minimize the transportation cost for the heterogeneous vehicle routing problem with two-dimensional loading constraints. A simulated annealing algorithm combined with heuristic local search is proposed, and a set of bin loading heuristic algorithms are designed to solve the bin loading problem. Zhang et al. [14] took the total cost minimization as the optimization objective, studied the heterogeneous vehicle routing problem with loading constraints, designed a hybrid particle swarm optimization algorithm combining artificial bee colony and artificial immune to solve the problem, and proposed several strategies to avoid the algorithm falling into local optimum. Sabar et al. [15] proposed a two-stage algorithm to solve the heterogeneous vehicle routing problem with two-dimensional bin loading constraints. In the first stage, an adaptive meme algorithm was designed to solve the routing problem. In the second stage, a hybrid algorithm based on five heuristic algorithms was used to solve the bin loading problem. Dominguez et al. [16] proposed a hybrid algorithm based on neighborhood search and loading heuristic algorithm to solve the two-dimensional vehicle routing problem with cluster backhaul and used randomization technology to guide the search process. Good results were obtained. Wang et al. [17] established a mathematical model for the logistics distribution problem with two-dimensional loading and proposed a memetic algorithm to solve the problem. The algorithm integrates the heuristic search knowledge of loading and route and improves the performance and result quality of existing algorithms. Literature research shows that the research on 2L-OVRPTW is still in its infancy.

The 2L-OVRPTW can establish two different problem models. One is the 0-1 mixed integer model (MIP) and the other is the order model. For MIP, the most exact algorithms such as branch and bound and column generation mainly solve the problem. The solution space can be searched through commercial solvers (such as GUROBI or CPLEX) for small-scale problems. The optimal solution can be found reasonably, but the exact algorithm will show a geometric growth trend with the increase of the problem size for large-scale problems. For the order model, it is usually composed of customer sequence of vehicle service. The constraints of such problems are often hidden in the order model, mainly by heuristic algorithm and intelligent algorithm. The heuristic algorithm is usually based on the characteristics of the problem to construct feasible solutions, such as Johnson algorithm, palmar algorithm, and Gupta algorithm. This kind of algorithm can obtain a feasible solution quickly, but it cannot guarantee that the solution is the best solution in the global range. Intelligent algorithm is a combination of the characteristics of the problem and intelligent mechanism to solve; through simulating some mechanisms in nature to guide the algorithm search, often in a short time, we can get a satisfactory solution or approximate optimal solution of complex scheduling problems. He pointed out in [18] that the effectiveness of intelligent algorithm depends on the model characteristics of scheduling problem and the mechanism of the intelligent algorithm. The intelligent optimization algorithm only needs to spend a short time searching part of the solution space of the scheduling model

to find a satisfactory solution so that it can effectively solve all kinds of complex scheduling problems.

The whale optimization algorithm (WOA) is a new swarm intelligence optimization algorithm proposed by Seyedali and Andrew [19]. WOA simulates the predator-prey process of humpback whales in nature and realizes the search of high-quality solution by searching, encircling, and attacking. WOA is widely used in many fields [20–23] because of its advantages of simple operation, few parameters, easy understanding, and strong optimization performance and has achieved good results. From the current research status of WOA, reasonable selection of adaptive weights in the search process can effectively control the search direction and further improve the search ability of the algorithm. At present, WOA is mainly used to solve continuous optimization problems, but for discrete optimization problems, the real solution space needs to be discretized. WOA has a certain effect on global search, but it cannot avoid the problem that the high-quality block structure is destroyed in the process of generating new species group. Therefore, WOA is used to construct a three-dimensional matrix after the first stage search in the solution space and to search in the second stage. The three-dimensional matrix is used to learn and accumulate the block structure and its location information based on customer service order. Then, the population updating process is guided to improve the search efficiency of learning whale optimization algorithm (LWOA), for example, for the customer service order of high-quality solution [1,3,5,2,4] and [2,4,5,1,3], where [2, 4] is a block structure based on the order relationship of customer 2 before customer 4 and [2, 4] appears in both the fourth position of the first solution and the first position of the second solution. Therefore, the three-dimensional matrix can retain the block structure of the first two and the last four and also record the location information of the block structure, that is, the block structure [2, 4] can be stored in the elements with subscripts (4,2,4) and (1,2,4), respectively, in the three-dimensional matrix (the first dimension subscript in the three-dimensional matrix represents the location information, and the second and third dimensions represent the order relationship of customers).

In this paper, the model and solution of 2L-OVRPTW are studied. In the model aspect, the 2L-OVRPTW model integrated by OVRPTW and BPP is established, and the optimization objective of this problem model is to minimize the driving distance. In the process of solving, through the analysis of the properties of the problem, a learning whale optimization algorithm (LWOA) is proposed to optimal vehicle routing problem, and an improved skyline method is designed to solve the loading problem. Specifically, three rules are used to initialize the population, and LOV rules are used to discretize the real solution. Secondly, in the global search, the whale optimization algorithm is used to find the optimal solution in the first stage. In the second stage, the block structure based on the order relation and the three-dimensional matrix of its position information are designed to effectively learn the solution and accumulate relevant information. Finally, the local search which combines the exchange operation, swap operation, and insert operation is

introduced to carry out the efficient and detailed local search for the solution obtained from the two-stage global search, which can effectively balance the global and local search of LWOA and improve the overall search ability of the algorithm. Simulation results and algorithm comparison show that LWOA is an effective algorithm for 2L-OVRPTW.

The rest of this paper is arranged as follows. In Section 2, the definition of the OVRPTW is introduced and a corresponding integer programming model is also proposed. In Section 3, the LWOA algorithm is proposed and described in detail. In Section 4, experimental results and comparisons are given and discussed. The current work is summarized and the future research direction is prospected in Section 5.

2. Two-Dimensional Loading Open Vehicle Routing Problem with Time Window

2.1. Problem Description and Symbol Definition of the 2L-OVRPTW. The 2L-OVRPTW can be described as follows. In directed network graph $G = (V, E)$, $V = \{0, 1, 2, \dots, N\}$ represents the vertex set, where vertex 0 represents central depot and $V' = V \setminus \{0\}$ represents the customer set. $E = \{(i, j) | i, j \in V, i \neq j\}$ is the arc set, and d_{ij} is the distance from customer i to customer j . There are N_k homogeneous vehicles in the central depot for service, the capacity of each vehicle is Q , and the length and width of the carriage are (L, W) ; the carriage area is $S = L * W$. M_i is the total number of rectangular goods required by customer i , q_i is the total weight of goods required by customer i , and q_{ik} is the weight of the k th item required by customer i ; then, $q_i = \sum_{k=1}^{m_i} q_{ik}$. M_{ig} is the g goods required by customer i , l_{ig} is the length of the g goods required by customer i , and w_{ig} is the width of the g goods required by customer i ; then, $s_{ig} = l_{ig} \cdot w_{ig}$. At the same time, considering the needs of customer satisfaction, the time window for customer i to receive service is set as $[e_i, h_i]$, e_i means the earliest service time allowed, h_i means the latest service time, and the service time of each customer is T_i . If the service starts within the time window of customer acceptance, the satisfaction is 1, otherwise it is 0. The time to arrive at the customer is st_i . If the service arrives in advance, it needs to wait until the service start time.

The objective of 2L-OVRPTW is to minimize the distance and consider the following constraints:

- (1) Each vehicle starts from the central depot and does not need to return back after serving all customers
- (2) When serving current customers, it is not necessary to move the goods of other customers (the principle of last in first out)
- (3) The loading edge of all articles must be parallel to the carriage edge
- (4) All goods in the carriage shall not be overlapped
- (5) The goods loaded on each vehicle shall not exceed the length and width of the carriage and the maximum load of the vehicle
- (6) Each vehicle must be serviced after the service time allowed by the customer

Figure 1 shows a distribution route of the 2L-OVRPTW. It can be seen from Figure 1 that three homogeneous vehicles serve 11 customers in the central depot. The service routes of vehicle 1, vehicle 2, and vehicle 3 are $route[1] = 0, 1, 2, 3$, $route[2] = 0, 4, 5, 6, 7$, and $route[3] = 0, 8, 9, 10, 11$. The loading scheme of vehicle 2 is shown in Figure 2. The goods are placed at the lower-left corner of the top view of the carriage (the front of the carriage is on the left). When unloading the goods, the current customer's goods are unloaded through a straight line parallel to the length of the carriage. The thick line in the figure represents the skyline, the thick line parallel to the length of the carriage is the longitudinal skyline, and the thick line parallel to the width of the carriage is the transverse skyline; the shaded part is the redundant space (wasted space) of the vehicle.

2.2. Mathematical Model of the 2L-OVRPTW

$$\min f_1 = \sum_{k=1}^{N_k} \sum_{i,j \in v} d_{ij} x_{kij}, \quad (1)$$

$$\text{s.t. } \sum_{i=1}^N q_i y_{ik} \leq Q, \quad k \in \{0, 1, \dots, N_k\}, \quad (2)$$

$$\sum_{k=1}^{N_k} y_{ik} = 1, \quad \forall i \in V', \quad (3)$$

$$\sum_{i=1}^N x_{ijk} = y_{ik}, \quad \forall j \in V', k \in \{0, 1, \dots, N_k\}, \quad (4)$$

$$\sum_{j=1}^N x_{ijk} = y_{ik}, \quad \forall i \in V', k \in \{0, 1, \dots, N_k\}, \quad (5)$$

$$0 \leq v_{kig} \leq W - w_{kig}, \quad k \in \{0, 1, \dots, N_k\}, \forall i \in V', \forall g \in M_i, \quad (6)$$

$$0 \leq h_{kig} \leq L - l_{kig}, \quad k \in \{0, 1, \dots, N_k\}, \forall i \in V', \forall l \in M_i, \quad (7)$$

$$\begin{aligned} HH_{ii'gg'} = & \left(\max(y_{ik} \cdot (H_{kig} + w_{ig}), y_{i'k'} \cdot (H_{ki'g'} + w_{i'g'})) \right. \\ & \left. - (y_{ik} \cdot w_{ig} + y_{i'k'} \cdot w_{i'g'}) \right), \\ & \forall i, i' \in V', \forall g, g' \in M_i, g \neq g', k \in \{0, 1, \dots, N_k\}, \end{aligned} \quad (8)$$

$$\begin{aligned} LL_{ii'gg'} = & \left(\max(y_{ik} \cdot (V_{kig} + l_{ig}), y_{i'k'} \cdot (V_{ki'g'} + l_{i'g'})) \right. \\ & \left. - (y_{ik} \cdot l_{ig} + y_{i'k'} \cdot l_{i'g'}) \right), \\ & \forall i, i' \in V', \forall g, g' \in M_i, g \neq g', k \in \{0, 1, \dots, N_k\}, \end{aligned} \quad (9)$$

$$\max(HH_{ii'gg'}, LL_{ii'gg'}) \geq 0, \quad \forall i, i' \in V', \forall g, g' \in M_i, \quad (10)$$

$$\begin{aligned} \max(HH_{ii'gg'}, (H_{kig} - H_{ki'g'})) \geq 0, \\ \forall i, i' \in V', \forall g, g' \in M_i, g \neq g', \end{aligned} \quad (11)$$

$$y_{ki} \cdot st_i \geq y_{ki} \cdot e_i, \quad \forall i \in V', k \in \{0, 1, \dots, N_k\}, \quad (12)$$

$$e_i \leq y_{ki} \cdot st_i + T_i \leq h_i, \quad \forall i \in V', k \in \{0, 1, \dots, N_k\}, \quad (13)$$

$$x_{ijk} = \begin{cases} 1, & \text{if customer } i \text{ is visited before customer } j \text{ by the vehicle } k, \\ 0, & \text{otherwise,} \end{cases} \quad (14)$$

$$y_{ijk} = \begin{cases} 1, & \text{if customer } i \text{ is visited by the vehicle } k, \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

In the above model, (1) gives the objective function of 2L-OVRPTW. Constraint (1) represents the minimization of the total distance. Constraint (2) ensures that the load of the vehicle does not exceed the maximum load of the vehicle. Constraint (3) ensures that every customer can be served. Constraints (4) and (5) ensure that every customer has and only has one vehicle to serve. Constraints (6)–(11) are the loading constraints, in which constraints (6) and (7) ensure that all items are fully loaded into the compartment. Whereas w_{kig} and l_{kig} , respectively, represent the g item width and length of the k vehicle serving the i customer. H_{kig} and V_{kig} , respectively, represent the horizontal and vertical coordinates of the upper-left corner of the g item of the i customer served by the k vehicle. Constraints (8)–(10) show that customers cannot overlap. Constraint (11) satisfies the principle of first in and last out. Constraints (12) and (13) show the time window constraints. Constraints (14) and (15) show the value range of decision variables.

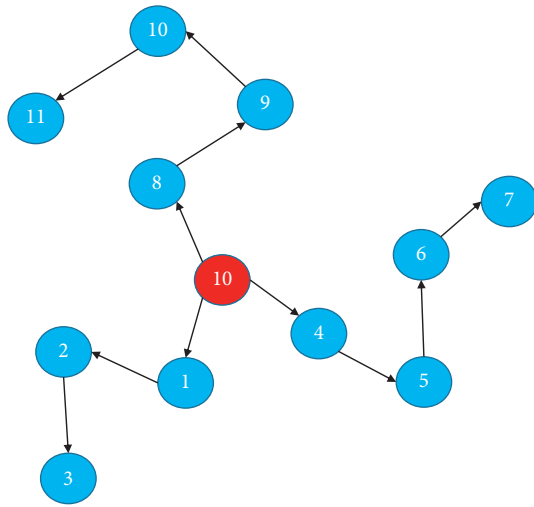
3. The LWOA for Solving 2L-OVRPTW

3.1. Global Search. In this section, the population initialization, standard WOA, block structure and three-dimensional matrix, local search, and loading strategy are introduced. Finally, the loading flowchart and the flowchart of LWOA are given.

3.1.1. Population initialization. In order to ensure the diversity and dispersion of the solutions, two individuals are generated by **rule1** and **rule2**, and then, the rest individuals are generated by **rule3**. **Rule1** considers the nearest neighbor principle, **rule2** considers the highest satisfaction principle, and **rule3** adopts the quasi-opposition strategy. Placing all unserved customers in set Z , the specific rules are as follows:

Rule1 is the nearest neighbor principle.

After the vehicle departs from the central depot, the distance between the central depot and each customer



$\pi^{gen,w} = 0,1,2,3,0,4,5,6,7,0,8,9,10,11$
 route1 = 0,1,2,3
 route2 = 0,4,5,6,7
 route3 = 0,8,9,10,11

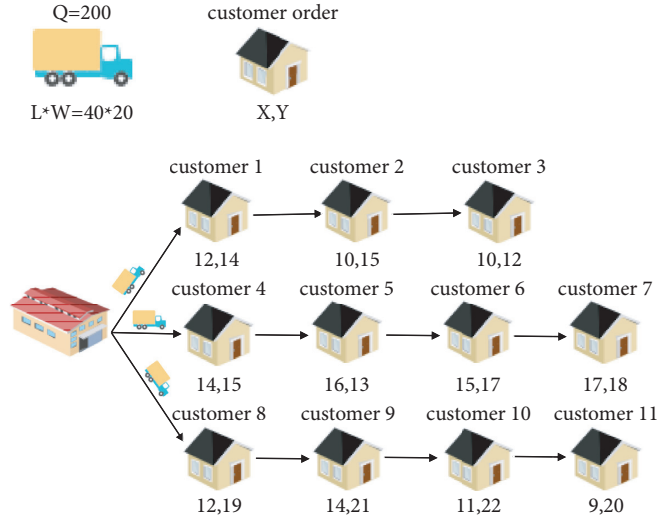


FIGURE 1: Distribution route of 2L-OVRPTW.

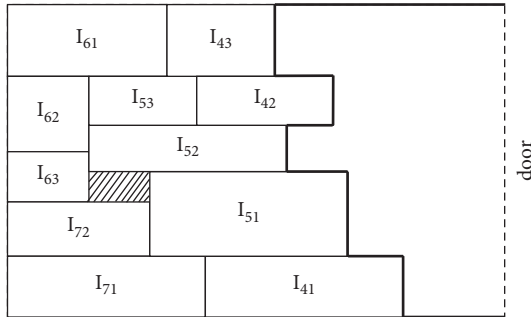


FIGURE 2: Vehicle loading scheme.

in Z is calculated, and the nearest customer i (if there are multiple customers, the customer with the highest satisfaction) is selected for service. The customer i from set Z was removed and the distances between customer i and other customers in Z are calculated. After selecting the nearest customer j , if the constraints are still met when continuing to serve the customer j , it is necessary to remove the customer j from set Z and calculate the distance between the customer and other customers in Z . Otherwise, the vehicle will give up serving the customer and dispatch the vehicle from the central depot to serve the customers in Z again. This cycle will last until all customers in Z are served.

Rule2 is the maximum satisfaction principle.

After vehicle starts from the central warehouse, we calculate the arrival time of each customer in Z from the central depot and select the customer i meeting the time window constraint to serve. If there are multiple customers whose arrival time is within the time

window, the nearest customer i is selected to serve. Move customer i out of set Z and calculate the time from customer i to other customers in Z . Select customer j that meets the time window constraint. If customer j still meets the constraint, it is necessary to move customer j out of set Z and calculate the arrival times from customer Z to other customers in Z . Otherwise, the vehicle will give up serving the customer and send the vehicle from the central depot to serve the customers in Z again until all the customers in Z are served.

Rule3 is quasi-antagonistic strategy.

Let the population size be $popsize$ and use the quasi-opposition strategy to generate $popsize-2$ individuals with n customers (that is, the search space is n). Let $\pi_i^{gen} = [\pi_i^{gen}[1], \pi_i^{gen}[2], \dots, \pi_i^{gen}[g], \dots, \pi_i^{gen}[S]]$ ($\pi_i^{gen}[g] \in V, n+1 \leq S \leq 2n$) denote their individual (distribution scheme) in the gen population. Let $P_i^{gen} = [P_i^{gen}[1], P_i^{gen}[2], \dots, P_i^{gen}[g], \dots, P_i^{gen}[n]]$ ($P_i^{gen}[g] \in V'$) be the customer-based discrete sort in individual π_i^{gen} (i.e., the customer sequence without central depot), and let $X_i^{gen} = [X_i^{gen}[1], X_i^{gen}[2], \dots, X_i^{gen}[g], \dots, X_i^{gen}[n]]$ ($X_i^{gen}[g] \in [a, b]$) be the real number sequence corresponding to the customer's discrete sort P_i^{gen} . As shown in equation (16), the quasi-antagonistic strategy is to obtain the opposite inverse solution (quasi-antagonistic solution) $X_j^{gen} = [X_j^{gen}[1], X_j^{gen}[2], \dots, X_j^{gen}[g], \dots, X_j^{gen}[n]]$ ($X_j^{gen}[g] \in [a, b]$) of the gen generation individual i through the calculation of equation (16) in the real number field $[a, b]$. Therefore, the population with quasi-opposite solution

can be obtained when the population is randomly generated. *popsi*-2 individuals are selected from the two populations as the next generation population:

$$X_j^{gen} [g] = \begin{cases} rand(\bar{X}, X_i^{gen} [g]), & X_i^{gen} [g] > \bar{X}, \\ rand(X_i^{gen} [g], \bar{X}), & X_i^{gen} [g] \leq \bar{X}, \end{cases} \quad \bar{X} = \frac{a+b}{2}. \quad (16)$$

The population obtained by X_i^{gen} through **rule3** is a real number sequence over a continuous field. Therefore, this paper uses LOV (large-order value) rule to transform the individual real number sequence X_i^{gen} into discrete customer sequence P_i^{gen} .

The specific steps of LOV rules are as follows:

Step 1: we select the real number sequence $X_i^{gen} = [X_i^{gen} [1], X_i^{gen} [2], \dots, X_i^{gen} [g], \dots, X_i^{gen} [n]]$, arrange X_i^{gen} in the descending order to get X_i^{geh} , find the corresponding sequence number X_i^{gen} in $X_i^{geh} [g]$, and place it in $\varphi_i^{gen} [g]$ to get the intermediate sequence φ_i^{gen} .

Step 2: for discrete customers, the values in P_i^{gen} are obtained successively by the formula $P_i^{gen} [\varphi_i^{gen} [g]] = g$

As shown in Table 1, an example of LOV application is given. If the real number sequence of an individual is $X_i^{gen} = [2.30, 1.59, 2.17, 1.78, 2.06, 1.89, 0.88]$, the real number sequence after descending is $X_i^{geh} = [2.30, 2.17, 2.06, 1.89, 1.78, 1.59, 0.88]$. Through the above steps, the intermediate sequence $\varphi_i^{gen} = [1, 6, 2, 5, 3, 4, 7]$ and the customer's discretization sequence $P_i^{gen} = [1, 3, 5, 6, 4, 2, 7]$ are obtained.

By calling the LOV rule, we can realize the transformation from real sequence to discrete sequence. Similarly, if the customer's discrete sequence changes, we can adjust it by using the reverse largest order value (RLOV) rule. Therefore, the RLOV rule is used to real number the discrete customer sequence obtained by **rule1** and **rule2** to get the corresponding real number solution. The initialization population is composed of *popsi*-2 individuals generated by **rule3**.

3.1.2. Standard WOA. Wang et al. [17] designed a whale optimization algorithm (WOA) by studying the foraging behavior of humpback whales. WOA mainly includes three search mechanisms: encircling search mechanism, spiral search mechanism, and random search mechanism.

Encircling search mechanism: when finding prey (solution), individual whales communicate with each other in the whole population so that the individual whales in the population keep approaching the prey. The mathematical model of encircling search mechanism is shown in formula (17)~(18):

$$D = |CX^{gen*} - X_i^{gen}|, \quad (17)$$

$$X_i^{gen+1} = X^{gen*} - A \cdot D, \quad (18)$$

where *gen* is the current number of iterations, X_i^{gen} is the *i* individual in the *gen* generation, X^{gen*} is the position of prey in the *gen* generation, *A* and *C* are coefficient vectors, *a* is the control parameter, *t* is the current number of iterations, each iteration updates the population, and T_{max} is the maximum number of iterations, with the increase of the number of iterations.

Spiral search mechanism: after calculating the distance from the prey, the individual whale takes the prey as the center and constantly approaches it in a spiral form. The mathematical model of spiral search mechanism is shown as

$$X_i^{gen+1} = D \cdot e^{bl} \cos(2\pi l) + X_i^{gen}, \quad (19)$$

where *b* is the parameter controlling the shape of the spiral, *l* is the random value of $[-1, 1]$, and γ is the search factor.

In equation (20), the probability *p* is introduced to choose whether to update the position information of individual whale by using the encircling search mechanism or the spiral search mechanism:

$$X_i^{gen+1} = \begin{cases} X_i^{gen} - A \cdot D, & p < \gamma, \\ D \cdot e^{bl} \cos(2\pi l) + X^{gen*}, & p \geq \gamma. \end{cases} \quad (20)$$

Random search mechanism: when the value of *A* is greater than 1, it will search outside the bounding circle. When we cannot get the effective information of the prey, we will choose a random way to search the prey to update the position information of the population, which can effectively enhance the diversity of the population. Then, the mathematical model of random search mechanism is as follows:

$$X_i^{gen+1} = X^{gen'} - A \cdot D, \quad (21)$$

$$D = |CX^{gen'} - X_i^{gen}|, \quad (22)$$

where $X^{gen'}$ is a randomly selected individual in the *gen* generation population.

3.1.3. Block Structure and Three-Dimensional Matrix. For any solution of 2L-OVRPTW, the block structure refers to the order of two consecutive adjacent customers in the solution. In the process of updating, the solution set shows certain similarities in the structure of the solution (block structure and its location information). The three-dimensional matrix proposed in the second stage of global search can effectively learn and accumulate the block structure. The location information of the solution set is obtained in the first stage so as to improve the overall effect of the algorithm. Because the same block structure in different positions will have a great impact on the target value of the solution, the first dimension of the designed three-dimensional matrix is used to record the location information of the block structure, and the other two dimensions represent two customers in the block structure.

TABLE 1: Solution representation of LOV.

Dimension (g)	1	2	3	4	5	6	7
$X_i^{gen} [g]$	2.30	1.59	2.17	1.78	2.06	1.89	0.88
$\varphi_i^{gen} [g]$	1	6	2	5	3	4	7
$P_i^{gen} [g]$	1	3	5	6	4	2	7

We define $Best_P(gen) = \{Best_p^{gen,1}, Best_p^{gen,2}, \dots, Best_p^{gen,g}, \dots, Best_p^{gen,n}\}$, where $Best_P(gen)$ is the gen generation population, and $Best_p^{gen,w} = \{Best_p_1^{gen,w}, Best_p_2^{gen,w}, \dots, Best_p_g^{gen,w}, \dots, Best_p_n^{gen,w}\}$, where $Best_p^{gen,w}$ is the w individual and $Best_p_g^{gen,w}$ is the g customer in the w individual of gen population. Let $TM_{l \times n \times n}^{gen}$

represent the three-dimensional matrix based on block structure and its location information in gen generation population, where $TM_{l \times n \times n}^{gen}(p, x, y)$ represents the value of block structure (x, y) in p position and $PM_{l \times n \times n}^{gen}(p)$ is the two-dimensional probability matrix normalized at p position. The specific definition is as follows:

$$ONE_TM_{l \times n \times n}^{gen,w}(p, x, y) \begin{cases} 1, & x = Best_p_p^{gen,w}, y = Best_p_{p+1}^{gen,w}, \\ 0, & \text{else,} \end{cases} \quad p = 1, 2, \dots, l, x, y = 1, 2, \dots, n, \quad (23)$$

$$TM_{l \times n \times n}^{gen}(p, x, y) = \sum_{w=1}^{popsize} ONE_TM_{l \times n \times n}^{gen,w}(p, x, y), \quad p = 1, 2, \dots, l, x, y = 1, 2, \dots, n, \quad (24)$$

$$TM_{l \times n \times n}^{gen}(p, x) = [TM_{l \times n \times n}^{gen}(p, x, 1), TM_{l \times n \times n}^{gen}(p, x, 2), \dots, TM_{l \times n \times n}^{gen}(p, x, n)], \quad p = 1, 2, \dots, l, x = 1, 2, \dots, n, \quad (25)$$

$$TM_{l \times n \times n}^{gen}(p) = \begin{bmatrix} TM_{l \times n \times n}^{gen}(p, x, 1), \dots, TM_{l \times n \times n}^{gen}(p, x, n) \\ TM_{l \times n \times n}^{gen}(p, 1, 1), \dots, TM_{l \times n \times n}^{gen}(p, 1, n) \\ \vdots \\ TM_{l \times n \times n}^{gen}(p, n, 1), \dots, TM_{l \times n \times n}^{gen}(p, n, n) \end{bmatrix}, \quad (26)$$

$$TM_{l \times n \times n}^{gen} = [TM_{l \times n \times n}^{gen}(1), TM_{l \times n \times n}^{gen}(2), \dots, TM_{l \times n \times n}^{gen}(l)], \quad (27)$$

$$PM_{l \times n \times n}^{gen}(p) = \begin{bmatrix} PM_{l \times n \times n}^{gen}(p, x, 1), \dots, PM_{l \times n \times n}^{gen}(p, x, n) \\ PM_{l \times n \times n}^{gen}(p, 1, 1), \dots, PM_{l \times n \times n}^{gen}(p, 1, n) \\ \vdots \\ PM_{l \times n \times n}^{gen}(p, n, 1), \dots, PM_{l \times n \times n}^{gen}(p, n, n) \end{bmatrix}, \quad (28)$$

$$\sum_{x=1}^n \sum_{y=1}^n PM_{l \times n \times n}^{gen}(p, x, y) = 1, \quad p \in \{1, 2, \dots, l\}, \quad (29)$$

where $ONE_TM_{l \times n \times n}^{gen,w}(p, x, y)$ is the number of times that customer $x = Best_p_p^{gen,w}$ and customer $y = Best_p_{p+1}^{gen,w}$ appears in p position of individual w in the gen generation, that is, the number of times that block structure $[x, y]$ appears in p position of individual w . For example, for individual $Best_p^{gen,1} = [1, 3, 2, 5, 4, 6]$ and individual $Best_p^{gen,2} = [2, 5, 3, 6, 1, 4]$, let $p = 2$; then, the client block structure of individual 1 in position 2 is $[3, 2]$, and the client block structure of individual 2 in position 2 is $[5, 3]$. Therefore, $ONE_TM_{l \times n \times n}^{gen,1}(2, 3, 2) = 1$, $ONE_TM_{l \times n \times n}^{gen,2}(2, 3, 2) = 0$, $ONE_TM_{l \times n \times n}^{gen,1}(2, 5, 3) = 0$, and $ONE_TM_{l \times n \times n}^{gen,2}(2, 5, 3) = 1$.

As shown in Figure 3, the block structure information of each p position is recorded in $TM_{l \times n \times n}^{gen}(p)$.

In the first stage of global search, the improved WOA is used to search in the whole solution space to obtain high quality. In the second stage, a three-dimensional matrix TM is constructed based on the high-quality solution obtained in the first stage. When the population is updated by using the three-dimensional matrix, the structure and location information of customer blocks in the current high-quality solution is learned and accumulated after each update and used to generate new individuals. The specific update process is as follows:

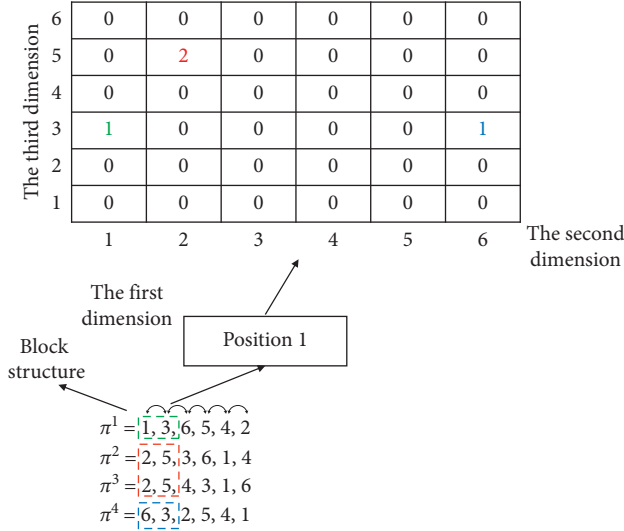


FIGURE 3: Three-dimensional matrix based on block structure.

Step 1: construct a three-dimensional probability matrix $TM_{l \times n \times n}^{gen}$ by using the high-quality solution obtained by WOA and construct a list $cos_sequence$ of remaining optional customers and a list $tabu$ of selected customers. Perform **step 2**.

Step 2: let $p = 1$, and randomly generate a random number $a, a \in (0, 1)$ and execute **step 3**.

Step 3: select the matrix of position p , set the occurrence times of block structure containing customers in $tabu$ to 0, and get the probability matrix of position p after $TM_{l \times n \times n}^{gen}(p)$ normalization as $PM_{l \times n \times n}^{gen}(p)$; $sum = \sum_{x=1}^n \sum_{y=1}^n PM_{l \times n \times n}^{gen}(p, x, y)$; if $sum > 0$, perform **step 4**; otherwise, perform **step 5**.

Step 4: if there is a selectable block structure, the roulette method is used to select the customer block structure for $PM_{l \times n \times n}^{gen}(p)$. Let $p = p + 2$; perform **step 6**.

Step 5: if there is no optional block structure in the current position, randomly select a customer in $cos_sequence$ and place it in this position; let $p = p + 1$, and then, go to **step 6**.

Step 6: delete the selected customers from $cos_sequence$ and add the selected customers to $tabu$.

Step 7: if $p > num_customer$, output the generated new individual $tabu$; if $p = num_customer$, return to **step 5**; otherwise, return to **step 3**.

The pseudocode of the LWOA algorithm is as follows.

The values range of γ , $popsize$, and max_l are analyzed in detail in Section 4.2.

3.2. Local Search. According to the characteristics of the 2L-OVRPTW problem, the solution space is huge, and the high-quality solutions are usually distributed in each local region of the solution space. Therefore, it is necessary to search the high-quality solutions from the global search; VNS includes three neighborhood operations: interchange operation,

insert operation, and swap operation. Interchange operation is used between vehicles, and insert operation and swap operation are used inside vehicles. In this way, local search can be carried out more carefully to update the current optimal solution. The variable neighborhood local search execution pseudocode is as follows.

3.3. Complexity Analysis of the LWOA. Let the population size be $popsize$, the number of running iterations of global search be gen_{global} , and the number of iterations of each generation of local search be gen_{local} . The number of times that local search actually uses neighborhood operators in each generation of search is $Z(Z = 20 * 40)$. The algorithm complexity of LWOA T_{LWOA} is composed of global search complexity and local search complexity, respectively. In each generation, the global search complexity consists of population initialization complexity $O(popsize * N)$, where N is the number of customers), discretization complexity ($2 * popsize * N!$), three-dimensional probability matrix update complexity $O(N^2 * (N - 1))$, complexity $O(popsize * (1 + N^2 * (N - 1)))$ of generating new population based on three-dimensional probability matrix, and population evaluation complexity ($popsize * N$); the result is $O(popsize * (1 + N^2 * (N - 1)))$. The complexity of local search is $O(gen_{local} * Z * (N))$ (the worst-case scenario).

Through the above analysis, $T_{LWOA} = O(gen_{global} * (popsize * (1 + N^2 * (N - 1)) + gen_{local} * Z * (N)))$.

3.4. Loading Strategy. For the loading problem based on the principle of “first in, last out” in the customer service order, this paper adopts the loading strategy of the skyline filling method to check every feasible solution generated. Only when the loading constraint is satisfied can the solution be retained. This section mainly includes initialization of loading order and loading mechanisms.

3.4.1. Initialize Loading Sequence. Each time we get a solution, we need to make sure that all the customers’ goods can be loaded into the car in the reverse order of the customers’ order. However, the loading order of multiple goods of each customer is not unique. Therefore, this paper proposes two rules for the initial loading order of customers’ goods.

Rule1: sort items according to their length

Rule2: sort items by area from small to large

For example, if the vehicle service sequence is customer 1, customer 2, customer 3, and customer 4, customer 4 of the final service uses **rule1** for initial loading and the other three customers use **rule2** for initial loading.

3.4.2. Loading Method. The loading process of articles is based on the skyline. Initially, the horizontal skyline and the vertical skylines are the width of the carriage and the length of both sides of the vehicle, respectively. The loading position of articles is determined by the score of articles placed on each section of the horizontal skyline.


```

(1) Input:
(2) Let the number of customers be  $N$ .
     $M_n$  represents the total number of items required by the customer  $n$ .
     $gen$  be the current number of iterations; the maximum number of iterations is  $gen_{max}$ .
     $\pi_i^{gen}$  is the individual  $i$  of generation  $gen$ , and the population size is  $popsize$ .
     $\gamma$  is the search factor.
    The update times of three-dimensional matrix is  $\max\_l$ .
    The vehicle number is  $k, k \in K$ .
     $I_{ng}$  indicates the item  $g$  of the customer  $n$ 
(3) Let  $A = 2ar_1 - a, C = 2r_2, r_1, r_2 \in \text{ran do } m[0, 1], a = 2 - 2 * gen/gen_{max}, b = 1, l \in \text{ran do } m[-1, 1], p \in \text{ran do } m[0, 1]$ 
(4) begin:
(5) Population initialization: using rule1 and rule2 in Section 3.1.1 to generate an individual and using rule3 to generate  $popsize - 2$  individuals.
(6) for each  $gen \in gen_{max}$  do
(7)   for each  $i \in popsize$  do
(8)     if  $A > 1$  then
(9)       update  $\pi_i^{gen}$  by equation (21)
(10)    else
(11)      update  $\pi_i^{gen}$  by equation (20)
(12)    end if
(13)  end for
(14) end for
(15) for each  $gen \in \max\_l$  do
(16)   Choosing the best individuals  $\pi^{gen}$  to construct the three-dimensional matrix
(17)   Update the population by the three-dimensional matrix
(18) end for
(19) Choosing the best individual as  $\pi_{best}^*$ 
(20) Loading initialization: Initialize the loading sequence of all customers using rules 1 and 2 in 3.4.1.
(21) for each  $k \in \pi_{best}^*$  do
(22)   for each  $n \in k$  do
(23)     for each  $I_{ng} \in n$  do
(24)       Loading the customer's goods into the carriage through the loading strategy in Section 3.4.2
(25)     end for
(26)   end for
(27) end for
(28) Output:
(29) return  $\pi_{best}^*$ 

```

ALGORITHM 1: Learning whale optimization algorithm.

(1) When the width is equal to the horizontal skyline, if the length is equal to either side of the adjacent vertical skyline, the score is 5. If the length is lower than the longitudinal skyline of the lowest adjacent side, the score is 4. If the length is between the two sides, the score is 3. Otherwise, the score is 2. (2) If the width is less than the transverse skyline, the score is 4 if the length is equal to either side of the longitudinal skyline of the two adjacent sides. If the length is lower than the longitudinal skyline of the lowest adjacent side, the score is 3. Otherwise, the score is 2. (3) If the width is greater than the transverse skyline, the score is 1. If all the horizontal skylines are less than the width of the object, it is necessary to fill the skyline before loading the object. In addition, when placing the object, it is also necessary to consider whether it exceeds the length and width limits of the vehicle. Loading the object into the vehicle and updating the skyline at the position with the highest score.

3.5. *Flowchart of LWOA.* This section gives the flowchart of LWOA. In the whole operation process, the individual needs to be boxed after each update of the individual, and the individual meeting the boxing constraints will be retained. The overall schematic diagram of the LWOA algorithm and the detailed flowchart of local search are shown in Figures 4 and 5.

4. Simulation Experiment and Comparative Analysis

The main content of this section includes the design of customer goods, performance index and parameter setting, and simulation experiment comparison and analysis. All experiments are run on Windows 10 platform, which has 3.0 GHz CPU and 8 GB RAM, and are on a single thread. The algorithm in this paper is implemented by Python 3.7.

```

(1) Input:
(2) Choose the best individual  $\pi_i$  and convert it to the permutation  $\pi_0$  according to the LOV rule. Perform a neighborhood operation on  $\pi_0$  to get  $\pi$ 
(3) begin:
(4) Set loop = 1, max-loop = 40,  $n = 1$ , max- $n = 20$ 
(5) while  $n < \text{max-}n$  do
(6) Random select  $m-k$ ,  $m-k \in \{1, 2, 3\}$ 
(7) Randomly select  $k$  and  $l$ , where  $k \neq l$ 
(8) if  $m-k = 1$ , then  $\pi' = \text{insert}(\pi, k, l)$ 
(9) else if  $m-k = 2$ , then  $\pi' = \text{swap}(\pi, k, l)$ 
(10) else if  $m-k = 3$ , then  $\pi' = \text{inverse}(\pi, k, l)$ 
(11) end if
(12) end if
(13) end if
(14) if  $\pi' < \pi$ , then  $\pi = \pi'$ 
(15) else  $n = n + 1$ 
(16) end if
(17) end while
(18)  $\pi' = \text{interchange}(\pi, k, l)$ 
(19) if  $\pi' < \pi$ , then  $\pi = \pi'$ 
(20) end if
(21) until loop = max-loop
(22) if  $\pi < \pi_0$ , then  $\pi_0 = \pi$ 
(23) end if
(24) Output:
(25) Convert  $\pi_0$  back to  $\pi_i$  according to the RLOV rule

```

ALGORITHM 2: Variable neighborhood local search.

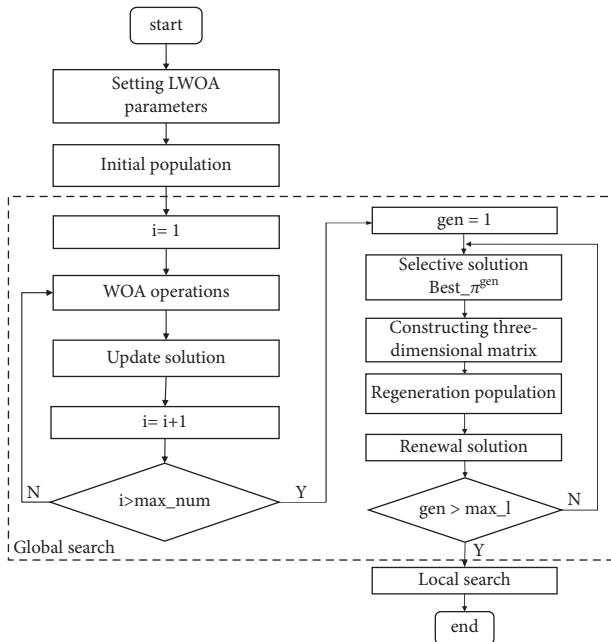


FIGURE 4: LWOA flowchart.

4.1. Experimental Setup. Since there is no standard testing benchmark suitable for 2L-OVRPTW, this paper uses the dataset and generates customer items according to [24], as shown in Table 2. In the homogeneous fleet, the length and width (L , W) of the car are (40, 20). Category 1 corresponds

to the pure CVRP instance, and no boxing constraint is imposed on it. Therefore, this paper uses categories 2–5 to generate rectangular items needed by customers. As shown in Table 2, select category 4 for the dataset with less than 25 customers, that is, randomly select the number of items required by customers in $[1, 4]$ and then assign one of three possible shapes to each item with the same probability. The three shapes are vertical (relative height is greater than relative width), uniform (relative height and width are generated in the same interval), or horizontal (relative height is less than relative width). Similarly, for datasets with more than 25 customers, category 3 is used.

4.2. Parameter Setting. The parameters involved in LWOA include population size, search factor, and learning times of the three-dimensional matrix. This paper selects problem C201 with 25 customers for experimental analysis and uses design of experiment (DOE) to explore the influence of parameters on LWOA performance. Four level values are selected for each parameter, as shown in Table 3, and the orthogonal test table of the L16 (4^3) scale is established. In each group of parameters, the algorithm runs independently 20 times, intending to minimize the mileage. The experimental results are as follows. The experimental results are as follows.

It can be seen from Tables 3–5 that different parameter combinations have a great influence on the algorithm. Therefore, to maximize the algorithm's performance, this paper sets the parameter combination based on the

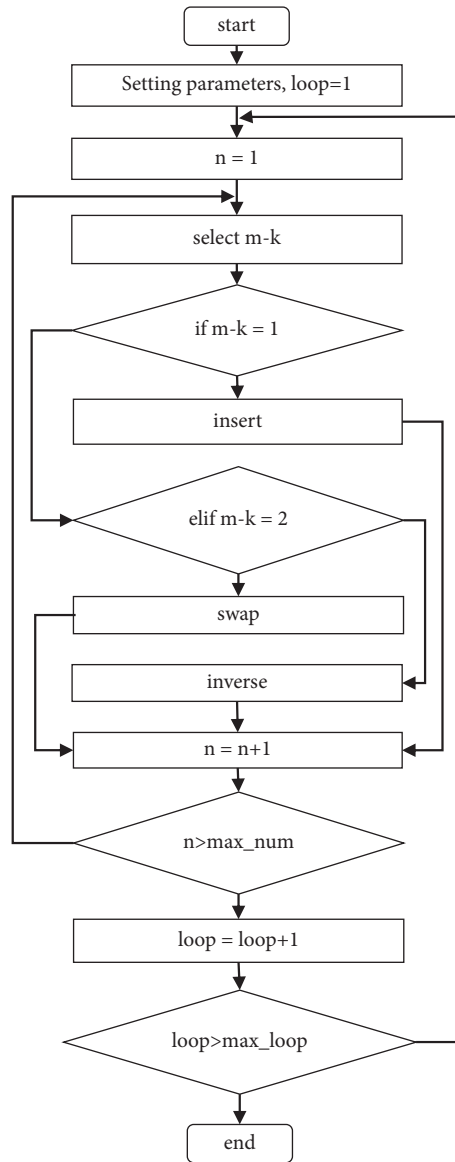


FIGURE 5: Local search flowchart.

TABLE 2: Item generation category.

Category	m_i	Vertical		Regular		Level	
		Relatively long	Relatively wide	Relatively long	Relatively wide	Relatively long	Relatively wide
2	[1, 2]	[0.4L, 0.9L]	[0.1W, 0.2W]	[0.2L, 0.5L]	[0.2W, 0.5W]	[0.1L, 0.2L]	[0.4W, 0.9W]
3	[1, 3]	[0.3L, 0.8L]	[0.1W, 0.2W]	[0.2L, 0.4L]	[0.2W, 0.4W]	[0.1L, 0.2L]	[0.3W, 0.8W]
4	[1, 4]	[0.2L, 0.7L]	[0.1W, 0.2W]	[0.1L, 0.4L]	[0.1W, 0.4W]	[0.1L, 0.2L]	[0.2W, 0.7W]
5	[1, 5]	[0.1L, 0.6L]	[0.1W, 0.2W]	[0.1L, 0.3L]	[0.1W, 0.3W]	[0.1L, 0.2L]	[0.1W, 0.6W]

experimental analysis. That is, the later experimental part will also be based on this parameter combination.

4.3. Comparison and Analysis of Simulation Experiments. In order to verify the effectiveness of LWOA in solving 2L-OVRPTW, this paper compares LWOA with WOA, GA, and

EDA. The objective of the optimization is to minimize the driving mileage for 20 independent experiments. The three performance indexes of optimal value (BST), the worst (WST) and the average value (AVG) are selected for comparative analysis, and the optimal results are roughened. The experimental results are shown in Table 6. It can be seen from Table 6 that the three indexes of LWOA in most cases

TABLE 3: Parameter level.

Parameter	Level			
	1	2	3	4
<i>popsize</i>	30	50	70	90
γ	0.2	0.4	0.6	0.8
<i>max_l</i>	20	40	60	80

TABLE 4: Results of orthogonal experiment.

Parameter	Level			AVE
	<i>popsize</i>	γ	<i>max_l</i>	
1	1	4	3	395.988
2	1	3	4	404.856
3	1	2	2	412.829
4	1	1	1	411.833
5	2	4	1	379.291
6	2	3	2	393.795
7	2	2	4	383.189
8	2	1	3	382.693
9	3	4	1	387.112
10	3	3	3	374.384
11	3	2	2	359.409
12	3	1	4	379.464
13	4	4	3	377.916
14	4	3	2	369.746
15	4	2	1	380.632
16	4	1	4	376.294

TABLE 5: Average response value of parameters.

Level	<i>popsize</i>	γ	<i>max_l</i>
1	406.376	390.176	389.717
2	384.742	384.015	383.945
3	375.092	385.695	382.745
4	377.569	385.077	387.763
Range	31.284	6.161	6.972
Grade	1	2	3

TABLE 6: Comparison of LWOA, WOA, GA, and EDA.

Problem	LWOA			WOA			GA			EDA		
	BST	WST	AVG	BST	WST	AVG	BST	WST	AVG	BST	WST	AVG
C101	242.01	311.58	279.80	245.30	327.05	288.46	259.00	333.53	294.73	243.28	323.21	288.33
C102	247.62	317.52	284.29	252.38	333.87	295.02	261.09	321.98	292.14	251.17	327.60	287.46
C103	224.83	315.95	279.87	252.77	326.05	283.24	252.57	354.41	290.81	246.63	341.71	286.53
C104	245.98	308.16	280.61	268.13	338.33	296.82	258.78	314.58	299.73	245.44	330.65	292.10
C105	242.79	301.49	280.94	234.54	311.92	277.52	254.55	333.01	289.48	253.74	326.57	283.90
C106	242.15	317.8	283.21	266.43	345.98	297.04	230.55	332.64	288.16	254.22	329.45	295.28
C107	248.19	306.76	279.81	261.62	310.61	283.85	248.30	316.79	282.72	255.93	328.17	294.70
C108	232.96	321.57	285.42	260.09	329.08	292.76	241.78	326.56	285.77	242.38	314.09	279.75
C109	224.48	318.68	279.85	253.77	338.80	287.91	236.75	330.6	294.60	247.37	339.38	288.42
C201	328.36	406.8	373.00	357.83	420.27	389.59	343.67	413.31	384.74	342.78	420.57	382.82
C202	348.97	410.72	385.56	352.52	420.19	389.23	332.24	425.89	386.71	353.85	446.74	392.55
C203	353.11	425.83	387.58	358.11	435.19	395.75	356.26	451.24	388.47	357.60	428.34	383.84
C204	337.79	415.66	379.23	354.86	431.27	384.55	294.66	448.28	384.45	337.60	456.18	399.28
C205	347.33	416.55	382.22	361.29	417.95	394.90	317.31	429.01	383.39	356.92	441.02	387.02
C206	337.47	409.87	388.76	364.22	423.06	395.74	365.30	433.86	391.15	363.18	436.91	395.05
C207	336.32	418.19	382.28	359.74	420.08	388.47	340.18	422.48	385.61	337.76	431.17	391.35
C208	339.42	427.92	376.57	346.28	425.14	381.89	340.07	446.34	386.12	340.42	438.14	394.69

TABLE 7: Results of statistical analysis.

Paired test	The value of P		
	BST	WST	AVG
LWOA and WOA	0.000846	0.000503	0.000420
LWOA and GA	0.227385	0.000293	0.000293
LWOA and EDA	0.000503	0.000421	0.001184

are significantly better than other comparison algorithms. This shows that the proposed three-dimensional matrix can effectively learn and accumulate block structure information based on customer order relationship, improve the overall search effect of the algorithm, and verify the effectiveness of LWOA in solving 2L-OVRPTW.

For the objective function values (BST, WST, and AVG) as test samples, nonparametric paired sample test is carried out on the operation results of each algorithm with 95% confidence. The results are shown in Table 7.

5. Conclusions

In this paper, a learning whale optimization algorithm (LWOA) combined with improved skyline filling algorithm (ISFA) is proposed to solve 2L-OVRPTW. The specific contributions are as follows. (1) Establish the 2L-OVRPTW model with the aim of minimizing the total distance. (2) In the route optimization stage, after the whale optimization algorithm is used to update the population, a three-dimensional matrix update strategy based on customer order relationship is designed to learn and accumulate the block structure and customer location information in the high-quality solution, which can effectively guide the population update. According to the characteristics of the problem, the variable neighborhood local search between and within vehicles is designed to carry out the detailed and efficient local search for the high-quality solution individuals after global search. (3) In the loading optimization stage, to improve the feasibility of the post solution of path optimization, ISFA is used to optimize the loading process, and the customer's goods are effectively loaded into the carriage through the scoring mechanism based on the skyline.

Future work will focus on the multiobjective problem with loading and time window constraints, refine the nature of the problem, and design an efficient algorithm to solve it so that the research is more in line with the actual distribution of logistics transportation.

Data Availability

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

Conflicts of Interest

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

Acknowledgments

This research was partially supported by the National Science Foundation of China (61963022), the National Science

Foundation of China (62173169), and National Science Foundation of China (51665025).

References

- [1] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 1, no. 6, pp. 80-81, 1959.
- [2] L. Ma, M. Huang, S. Yang, S. Yang, R. Wang, and X. Wang, "An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1-13, 2021.
- [3] M. Iori, J. J. S. González, and D. Vigo, "An exact approach for the vehicle routing problem with two-dimensional loading constraints," *Transportation Science*, vol. 41, no. 2, pp. 253-264, 2007.
- [4] E. E. Zachariadis, C. D. Tarantilis, and C. T. Kiranoudis, "A guided tabu search for the vehicle routing problem with two-dimensional loading constraints," *European Journal of Operational Research*, vol. 195, no. 3, pp. 729-743, 2009.
- [5] G. Fuellerer, K. F. Doerner, R. F. Hartl, and M. Iori, "Ant colony optimization of the two-dimensional loading vehicle routing problem," *Computers & Operations Research*, vol. 36, no. 3, pp. 655-673, 2009.
- [6] C. Duhamel, P. Lacomme, A. Quilliot, and H. Toussaint, "A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem," *Computers & Operations Research*, vol. 38, no. 3, pp. 617-640, 2011.
- [7] D. Guimarans, O. Dominguez, J. Panadero, and A. A. Juan, "A simheuristic approach for the two-dimensional vehicle routing problem with stochastic travel times," *Simulation Modelling Practice and Theory*, vol. 89, pp. 1-14, 2018.
- [8] L. Wei, Z. Zhang, D. Zhang, and A. Lim, "A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints," *European Journal of Operational Research*, vol. 243, no. 3, pp. 798-814, 2015.
- [9] S. Leung, X. Zhou, D. Zhang, and J. Zheng, "Extended guided tabu search and a new loading algorithm for the two-dimensional loading vehicle routing problem," *Computers & Operations Research*, vol. 38, no. 1, pp. 205-215, 2010.
- [10] L. Wei, Z. Zhang, D. Zhang, and S. Leung, "A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints," *European Journal of Operational Research*, vol. 265, no. 3, pp. 843-859, 2018.
- [11] Y. Rui, X. Zhu, Q. Zhang, Y. Qi, and Y. Lin, "Research of the model and algorithm for two-dimensional multi-depots capacitated vehicle routing problem with time window constrain," *Chinese Journal of Management Science*, vol. 25, no. 7, pp. 67-77, 2017.
- [12] Z. Wang and L. Zhou, "Modeling and optimization of vehicle scheduling problem with two-dimensional loading constraints," *Computer Technology and Development*, vol. 28, no. 10, pp. 105-110, 2018.

- [13] S. C. H. Leung, Z. Zhang, D. Zhang, X. Hua, and M. K. Lim, "A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints," *European Journal of Operational Research*, vol. 225, no. 2, pp. 199–210, 2013.
- [14] D. Zhang, Y. W. Si, F. Ye, and Q. Cai, "A hybrid swarm algorithm based on ABC and AIS for 2L-HFCVRP," *Applied Soft Computing*, vol. 64, pp. 468–479, 2018.
- [15] N. R. Sabar, A. Bhaskar, E. Chung, A. Turky, and A. Song, "An adaptive memetic approach for heterogeneous vehicle routing problems with two-dimensional loading constraints," *Swarm and Evolutionary Computation*, vol. 58, 2020.
- [16] O. Dominguez, D. Guimarans, A. A. Juan, and D. Ignacio, "A biased-randomised large neighbourhood search for the two-dimensional vehicle routing problem with backhauls," *European Journal of Operational Research*, vol. 255, no. 2, pp. 442–462, 2016.
- [17] Z. Wang, X. Hu, and X. Wang, "Vehicle routing problem in distribution with two-dimensional loading constraint," *Systems Engineering-Theory & Practice*, vol. 31, no. 12, pp. 2328–2341, 2011.
- [18] Y. J. He, B. Qian, and R. Hu, "Hybrid discrete teaching-learning-based optimization algorithm for solving complex parallel machine scheduling problem," *Acta Automatica Sinica*, vol. 46, no. 4, pp. 805–819, 2020.
- [19] M. Seyedali and L. Andrew, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, no. 1, pp. 51–67, 2016.
- [20] Y. Cai and P. Du, "Path planning of unmanned ground vehicle based on balanced whale optimization algorithm," *Control and Decision*, vol. 36, no. 11, pp. 2647–2655, 2021.
- [21] X. Yue, X. Peng, and L. Lin, "Short-term wind power forecasting based on whales optimization algorithm and support vector machine," *Proceedings of the CSU-EPSCA*, vol. 32, no. 2, pp. 146–150, 2020.
- [22] W. Zheng, Z. Li, H. Jia, and C. Gao, "Research on prediction model of steelmaking end point based on LWOA and LSSVM," *Acta Electronica Sinica*, vol. 47, no. 3, pp. 700–706, 2019.
- [23] K. Pankaja and V. Suma, "Plant leaf recognition and classification based on the whale optimization algorithm (WOA) and random forest (RF)," *Journal of The Institution of Engineers (India) Series B*, vol. 101, no. 2, pp. 597–607, 2020.
- [24] M. Gendreau, M. Iori, G. Laporte, and S. Martello, "A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints," *Networks*, vol. 51, no. 1, pp. 4–18, 2008.