

Article

Research on Algorithm for Multi-Objective Symmetric Collaborative Optimization Vehicle Routing Problem with Time Windows

Yipeng Zhang ^{1,2}, Junya Yang ¹, Pengyu Chen ^{2,*}  and Guoning Zhao ²

¹ CCCC Second Harbor Engineering Company Ltd., Wuhan 430014, China; zhangyipeng1@ccccltd.cn (Y.Z.); yangjunya@ccccltd.cn (J.Y.)

² School of Automotive Engineering, Wuhan University of Technology, Wuhan 430070, China; zgn@whut.edu.cn

* Correspondence: chen-2020@whut.edu.cn

Abstract

This paper proposes a collaborative optimization method that combines the multi-objective grey wolf optimizer (MOGWO) and the dual-thread ant colony optimization (DTACO) algorithm to solve the vehicle routing problem with time windows (VRPTW). The aim is to simultaneously optimize two often conflicting objectives: minimizing the number of vehicles and travel costs. Traditional methods usually optimize the number of vehicles or travel costs separately, making it difficult to balance the two and leading to insufficient resource utilization or excessively high travel costs in practical applications. To address this issue, this paper presents a symmetrical collaborative framework that integrates multi-objective grey wolf optimization with the dual-thread ant colony system to achieve synchronous and balanced optimization of multiple objectives. Moreover, to solve the problem of premature convergence of the algorithm, a backtracking mechanism is proposed, and its effectiveness is verified through ablation experiments. Experimental results show that this method significantly outperforms single-objective optimization algorithms on the Solomon dataset. Compared with the best known solutions (BKSs), this paper finds better solutions in some datasets, and the performance on the remaining datasets is also close to BKSs. For example, in C204-50, the lowest cost is reduced by 0.48%, in R102-50, the lowest cost is reduced by 1.29% and the number of vehicles is reduced by 1, and in RC106-50, the lowest cost is reduced by 8.48% and the number of vehicles is also reduced by 1. Therefore, the proposed algorithm provides an efficient meta-heuristic framework for VRPTW, effectively balancing the dual objectives and highlighting the value of symmetrical collaboration in multi-objective optimization.

Keywords: vehicle routing problem; time window constraint; multi-objective optimization; symmetry-inspired optimization; Pareto optimality



Received: 20 October 2025

Revised: 7 November 2025

Accepted: 11 November 2025

Published: 13 November 2025

Citation: Zhang, Y.; Yang, J.; Chen, P.; Zhao, G. Research on Algorithm for Multi-Objective Symmetric Collaborative Optimization Vehicle Routing Problem with Time Windows. *Symmetry* **2025**, *17*, 1949. <https://doi.org/10.3390/sym17111949>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the vigorous development of e-commerce and the rapid growth of on-demand delivery requirements, the importance of vehicle routing optimization problems in modern logistics systems has become increasingly prominent. As a classic combinatorial optimization problem in operations research, the vehicle routing problem was first proposed by Dantzig and Ramser [1] in 1959 and has since been a research hotspot in both academia and industry. In particular, the vehicle routing problem with time windows, which introduces strict time constraints based on traditional route optimization, is more in line

with the demands of actual logistic scenarios, such as fresh food delivery, medical supplies transportation and instant express services. Against this backdrop, VRPTW has emerged and has gradually become an important branch of VRP research. VRPTW requires optimizing vehicle scheduling plans under the premise of meeting the specified customer time windows, with the aim of minimizing travel costs, reducing the number of vehicles used, and improving customer satisfaction. However, VRPTW is an NP-Hard problem [2], and its difficulty in solving it increases exponentially with the scale of the problem. It is difficult to use traditional exact algorithms (such as branch and bound, dynamic programming) to solve large-scale problems efficiently. Therefore, designing efficient heuristic or meta-heuristic algorithms has become key to solving the VRPTW problem.

In recent years, metaheuristic algorithms have made breakthrough progress in solving VRPTW, demonstrating optimization performance that is difficult to match with traditional methods. Intelligent optimization methods represented by Ant Colony Optimization [3] and Grey Wolf Optimization [4] provide innovative solutions for complex logistics scheduling through unique biological behavior simulation mechanisms. The ACO algorithm draws on the principle of positive feedback from pheromones in the ant foraging process to achieve self-organizing optimization effects in the construction of the path, while the GWO algorithm, by simulating the social hierarchy and cooperative hunting strategies of grey wolf packs, exhibits outstanding global exploration capabilities. Despite the advantages of these algorithms, current research still faces several key challenges, including the following issues:

1. The limitation of single-objective optimization: Most algorithms adopt a single-objective optimization strategy, making it difficult to simultaneously consider the two conflicting objectives of number of vehicles and travel cost. In actual logistic operations, enterprises hope to minimize number of vehicles to reduce fixed costs while optimizing travel routes to save fuel and labor costs. This requires algorithms to have the ability to perform multi-objective collaborative optimization.
2. Imbalance between exploration and exploitation: Traditional algorithms often struggle to strike a balance between exploration and exploitation. Overemphasis on global exploration leads to slow convergence, while excessive local exploitation can easily result in premature convergence, especially when dealing with large-scale VRPTW problems.
3. Poor quality of initial solutions: The quality of the initial solution directly affects the convergence speed and the quality of the final solution. Randomly generated initial solutions often contain a large number of infeasible solutions, requiring significant computational resources for correction, which seriously affects the efficiency of the algorithm.

To address these challenges, this paper proposes a dual-threaded ACO that integrates multi-objective grey wolf optimization initialization. This algorithm efficiently solves VRPTW problems through a multi-stage collaborative optimization framework.

The MOGWO algorithm achieves collaborative optimization of multiple objectives such as travel cost and number of vehicles by introducing an adaptive weight mechanism and an elite retention strategy. This algorithm integrates the bionics principles of the social hierarchy and search behavior of grey wolves. Through the hierarchical guidance of α , β , and δ wolves, the population explores different directions towards the Pareto frontier, ultimately generating a set of uniformly distributed non-dominated solutions. Compared with the traditional Non-dominated Sorting Genetic Algorithm II [5], the MOGWO algorithm demonstrates stronger convergence performance and solution diversity when dealing with high-dimensional objective VRPTW problems.

On the other hand, the DTACO algorithm introduces the concept of “target balance symmetry” and employs two parallel ant colony systems to deeply explore different regions of the solution space. In the algorithm design, the main thread focuses on quickly generating feasible solutions to ensure solution efficiency, while the auxiliary thread refines the travel

cost under time window constraints. Both threads achieve a dynamic balance between global search and local exploitation through a pheromone sharing strategy, effectively overcoming the inherent defect of traditional ant colony algorithms that are prone to premature convergence. Moreover, the running time is significantly improved, with an average of 167.4 generations in the C dataset.

The main contributions of this paper are summarized as follows:

1. A novel hybrid adaptive framework integrating the MOGWO algorithm and LNS strategy is proposed. This framework introduces an adaptive weighting mechanism and a dynamic destruction–repair strategy, which can efficiently generate high-quality Pareto solutions for algorithm initialization. It effectively addresses the collaborative optimization challenge among multiple conflicting objectives and provides high-quality initial solutions after fifty iterations.
2. A DTACO algorithm with a differentiated LNS algorithm. To overcome the limitation of traditional methods focusing on a single objective, DTACO adopts a unique dual-thread architecture: the main thread focuses on travel cost optimization, while the auxiliary thread aims to reduce the number of vehicles. Through strategic pheromone sharing and constraint validation, this design ensures a dynamic balance between exploration and exploitation.
3. A novel solution backtracking mechanism to resolve optimization conflicts. It addresses the key issue of travel cost optimization being hindered by the rapid reduction in vehicle numbers in early iterations. The proposed backtracking mechanism, verified through ablation experiments, can effectively escape from such local optima, thereby significantly improving the quality and robustness of the overall solution.

The experimental part validates the effectiveness of the proposed algorithm based on the Solomon standard dataset. The results show that MOGWO-DTACO outperforms traditional methods in terms of the number of vehicles, travel costs, and computational efficiency, providing an efficient and balanced solution for complex logistics scheduling problems.

The structure of this paper is as follows: Section 2 reviews related research; Section 3 establishes the mathematical model of VRPTW; Section 4 details the design of the MOGWO-DTACO algorithm; Section 5 presents the experimental results and analysis; and Section 6 summarizes the paper and looks forward to future research directions.

2. Related Work

The VRPTW problem is different from the path planning problem. Simple single-objective and low-complexity algorithms such as Dijkstra’s algorithm [6], A* algorithm [7], RPT algorithm [8], branch and bound method [9], and dynamic programming method [10] are not applicable. To solve the path problem with time windows, the academic community has developed various heuristic and meta-heuristic algorithms such as simulated annealing algorithm [11], genetic algorithm [12], and particle swarm optimization algorithm [13], and applied them to the solution of VRPTW problems.

At present, VRP and its extended problems have formed a systematic theoretical system and method framework [14]. As an important variant of VRP, the VRPTW problems has become a benchmark problem for testing algorithm performance due to its strict temporal constraints. In response to the challenge of large-scale problems in VRPTW solution, the academic community has developed various heuristic and meta-heuristic algorithms.

Early research focused on smaller problem scales and mainly relied on exact algorithms. For example, Pecin et al. [15] proposed a branch-cut-and-price method, which significantly improved the solution efficiency of the capacitated vehicle routing problem. Yu et al. [16] introduced a multi-vehicle approximate dynamic programming algorithm and an integer branching method in the branch pricing algorithm to optimize carbon emissions during

the distribution process. However, the time window constraints significantly increase the complexity of the pricing sub-problem, and existing cutting planes are difficult to handle temporal conflicts. Therefore, these algorithms still face the challenge of high computational complexity when dealing with large-scale VRP problems with time window constraints.

To address this challenge, meta-heuristic algorithms have demonstrated significant advantages. Xie et al. [17] proposed a multi-strategy improved particle swarm optimization algorithm, which adopts a decreasing inertia weight strategy to balance global and local search, and introduces a random selection mechanism to enhance population diversity. Li et al. [18] proposed an improved genetic algorithm, which significantly improves the convergence speed and solution accuracy through an adaptive crossover and mutation mechanism and a reconstructed fitness function. However, single meta-heuristic algorithms have obvious limitations when solving VRPTW problems. When dealing with large-scale problems involving more than 100 customers, the error significantly increases, and it is difficult to balance exploration and exploitation capabilities, lacking an adaptive mechanism to handle dynamic constraints.

In terms of hybrid algorithm research, Exposito et al. [19] proposed a hybrid meta-heuristic algorithm combining the features of the greedy random adaptive search process and variable neighborhood search algorithm to solve the established mixed integer programming model. Guo et al. [20] proposed a genetic algorithm based on LNS, which significantly improves the solution quality by integrating ALNS and GA algorithms and designing new removal and reinsertion operators. These hybrid algorithm studies provide a theoretical basis for multi-strategy collaborative optimization, but still have significant deficiencies in adaptive mechanisms and multi-objective balance.

For the multi-objective optimization problem of VRPTW, existing research still has several key deficiencies. Ghannadpour et al. [21] designed a hybrid genetic algorithm for cash transfer to solve the constructed multi-objective game theory model, aiming to minimize the transfer risk and vehicle travel cost. Huang et al. [22] proposed a multi-objective hybrid optimization algorithm combining genetic algorithm, simulated annealing, and adaptive LNS. By using K-Means clustering for preprocessing and Dijkstra's shortest path calculation to construct the initial solution, it simultaneously minimizes travel cost and time window violations through multi-objective optimization strategies. These studies still have significant deficiencies in constraint satisfaction, solution set distribution quality, and scalability for large-scale problems.

In recent years, the grey wolf optimization algorithm and its multi-objective variants have demonstrated excellent convergence and solution quality in the VRPTW problem. Sun et al. [23] proposed an improved grey wolf optimization algorithm with dynamic disturbance coefficient and nonlinear control parameters, combined with a reverse learning strategy to enhance path planning performance. Chen et al. [24] proposed a hybrid multi-objective grey wolf algorithm. This algorithm achieves the conversion from continuous space to discrete paths through a novel encoding and decoding mechanism and integrates particle swarm optimization to enhance global search ability. The ant colony algorithm optimizes path construction by simulating the pheromone mechanism of ant colonies. Li et al. [25] proposed a multi-objective path planning method integrating ant colony and A* algorithms, significantly improving path smoothness and algorithm convergence speed by introducing a curvature suppression operator and an improved withdrawal mechanism. Wang et al. [26] designed a multi-objective simulated annealing-ant colony optimization algorithm to solve the constructed multi-objective model, aiming to minimize both vehicle travel costs and the number of vehicles used.

Compared with existing research, the MOGWO-DTACO algorithm proposed in this paper achieves breakthrough innovations in the following aspects: At the optimization mechanism level, it introduces an improved grey wolf optimization algorithm with non-dominated sorting strategy to achieve the collaborative optimization of vehicle number and

travel cost, solving the problem that traditional single-objective optimization is difficult to balance multi-objective conflicts; at the search strategy level, it introduces a dual-thread ant colony collaborative mechanism, where the global thread adopts a dynamic pheromone update strategy to enhance exploration ability, and the local thread combines an elite retention strategy to improve exploitation accuracy, effectively overcoming the imbalance between exploration and exploitation in traditional algorithms. At the local optimization level, the algorithm integrates a LNS operator based on destruction and repair, significantly enhancing its ability to escape from local optima. Experimental results show that this algorithm outperforms existing algorithms on C-type datasets in terms of solution sets.

3. Mathematical Model

3.1. Problem Description

The VRPTW problem is an extended model of the traditional vehicle routing problem with the addition of time window constraints for customer service. This problem requires planning the optimal routes for vehicles departing from a depot, ensuring that each customer is served within their specified time window while adhering to vehicle capacity limits. The time window constraints for customers can be either hard (service must be strictly within the time window) or soft (service outside the time window is allowed but incurs a penalty cost). The optimization objectives of VRPTW typically include minimizing travel costs, reducing the number of required vehicles, and lowering the cost of time window violations.

The network structure of this problem consists of a depot and multiple customer points. The Depot serves as the central node of the entire logistics system, responsible for centralized storage, sorting, and packaging of goods, and acts as the command and dispatch center of the entire distribution network. All delivery vehicles depart from the center after loading and must return to it after completing their delivery tasks. Each Customer has three key attribute features: load volume, coordinates, and time window constraints. The load volume determines the amount of goods that each vehicle needs to deliver to the customer point; the coordinates affect the spatial planning of the delivery routes; and most importantly, the strict time window constraints, including the earliest service time and the latest service time, which together form a service time interval that must be adhered to. These spatiotemporal constraints make the path planning problem more complex, requiring the optimization of vehicle routes under the condition of meeting all customer time window requirements to maximize delivery efficiency.

3.2. Mathematical Model and Constraints

In the VRPTW, the total operating cost is composed of both fixed costs of vehicles and variable costs of routes. Fixed costs arise from the basic expenses when vehicles are put into use and are directly related to the number of vehicles; variable costs, on the other hand, depend on the actual travel costs of vehicles, reflecting the dynamic consumption during the transportation process. Based on this, the following objective function for minimizing the total cost is established:

$$Z = \sum_{k \in K} C_f y_k + C_d \sum_{k \in K} \sum_{(i,j) \in A} d_{ij} x_{ijk} \quad (1)$$

Here, K represents the set of vehicles, A represents the set of arcs, d_{ij} represents the Euclidean distance from customer point i to j , x_{ijk} represents the decision variable (taking 1 when vehicle k passes through arc (i, j) , otherwise 0), y_k represents the 0-1 variable indicating whether vehicle k is used, C_f represents the fixed travel cost, and C_d represents the unit distance travel cost. Therefore, in the process of minimizing the total cost, the time window constraints and vehicle number constraints need to be considered.

The constraints include vehicle capacity constraints, time window constraints, path continuity constraints, and sub-loop avoidance constraints, which are specifically expressed as the following formulas.

1. Vehicle capacity constraints

To ensure that no vehicle is overloaded, the total cargo volume of each vehicle must not exceed its rated capacity:

$$\sum_{i=1}^N q_i \sum_{j=0}^N x_{ijk} \leq Q, \quad \forall k \in \{1, 2, \dots, K\} \quad (2)$$

Here, q_i represents the goods demand at customer point i , and Q represents the total carrying capacity of the vehicle. This constraint ensures that the total freight volume on any route does not exceed the vehicle's carrying limit by summing up the demand at each customer point served by each vehicle.

2. Time window constraint

The time window constraint is the core feature that distinguishes VRPTW from traditional VRP. The arrival time t_i of the vehicle at customer point i must be within its time window $[e_i, l_i]$:

$$e_i \leq t_i \leq l_i, \quad \forall i \in \{1, 2, \dots, N\} \quad (3)$$

Here, e_i and l_i respectively represent the earliest and latest service time of customer i . If the vehicle travels from customer point i to customer point j , its arrival time t_j needs to satisfy:

$$t_j = \max(t_i + s_i + d_{ij}, e_j) \quad (4)$$

Here, s_i represents the service time for customer i .

3. Path continuity constraint

To ensure that each customer is visited only once, the following constraint is established:

$$\sum_{k=1}^K \sum_{j=0}^N x_{ijk} = 1 \quad (5)$$

To ensure the closed nature of the vehicle routes, that is, the vehicles depart from the depot and return to the depot, the following constraints are established:

$$\sum_{j=1}^N x_{0jk} = 1, \quad \forall k \quad (6)$$

$$\sum_{i=1}^N x_{i0k} = 1, \quad \forall k \quad (7)$$

4. Avoiding sub-loop constraints

To prevent the occurrence of invalid sub-loops that do not include the depot in the solution, the following constraints are established:

$$u_i - u_j + Nx_{ijk} \leq N - 1, \quad \forall i, j \in \{1, 2, \dots, N\}, \forall k \quad (8)$$

Here, u_i and u_j are auxiliary parameters, representing the position of customer points i and j in the route. Through the above constraints, it is ensured that the VRPTW problem requires vehicles to depart from the depot, all customer points are visited exactly once,

the vehicle's cargo capacity is not less than the total demand of the customer points it serves, and the vehicle arrives at each customer point within the required time window.

3.3. Punishment Mechanism

In the process of solving the VRPTW problem, some solutions often fail to meet the constraints. Solutions that violate the constraints are handled through a penalty function, which mainly includes capacity penalties and time window violations. A dynamic penalty mechanism is also introduced in the penalty function, meaning that as the number of iterations increases, the dynamic weight of the penalty also increases to prevent incorrect paths from being repeatedly referenced, which could lead to the failure to find the optimal solution.

Penalty function:

$$F = D + \alpha(P_C + P_T) \quad (9)$$

Here, D represents the travel cost, P_C represents the penalty for vehicle overload, P_T represents the penalty for violating the time window, and α represents the dynamic penalty weight, which gradually increases with each iteration.

Capacity penalty:

$$P_C = \sum_{k=1}^K \max\left(0, \sum_{i=1}^N q_i y_{ik} - Q\right) \quad (10)$$

Here, $y_{ik} \in \{0, 1\}$ indicates whether vehicle k serves customer point i .

Time window penalty:

$$P_T = \sum_{i=1}^N [\max(0, e_i - t_i) + \max(0, t_i - l_i)] \quad (11)$$

Dynamic weighting: The penalty weight μ increases with the number of iterations to strengthen the constraints.

$$\mu = 100 \left(1 + \frac{\tau}{T_{\max}}\right) \quad (12)$$

Here, τ represents the current iteration number, and $T_{\max}$ is the maximum number of iterations.

4. Design of Dual-Threaded ACO Integrated with Multi-Objective Grey Wolf Optimization

4.1. Algorithm Framework

In view of the complexity of the VRPTW problem, as shown in Figure 1, this paper proposes a collaborative algorithm integrating multi-objective grey wolf optimization, LNS, and a dual-thread ant colony system based on the concept of "multi-objective symmetrical balance". This algorithm advances its optimization symmetrically under the time window constraints through a hierarchical optimization strategy, with the aid of multi-stage collaboration and dynamic coordination mechanisms, thereby significantly enhancing operational efficiency.

The algorithm begins with an initialization stage, where a node network and distance matrix are constructed to provide the necessary data foundation for subsequent path computations. Then, a greedy algorithm is used to generate an initial feasible solution, quickly constructing feasible paths through the "nearest neighbor" strategy. The initial solution is then passed to the multi-objective grey wolf optimization algorithm to enter the multi-objective grey wolf optimization stage.

In the multi-objective grey wolf optimization stage, the algorithm simulates the search behavior of wolves, guiding the population search through the leadership of α , β , and δ

wolves. This stage simultaneously optimizes the number of vehicles and travel cost, outputting a set of Pareto non-inferior solutions. To improve the quality of the solutions, a LNS strategy is embedded in this stage, employing a dynamic destruction–repair mechanism: controllable destruction is achieved by removing customer points by clusters, and intelligent repair is conducted based on a three-dimensional evaluation criterion guided by pheromones (integrating path segment continuity, distance cost, and time window urgency). Additionally, dual operators of node exchange and optimal insertion are applied to optimize the path structure, effectively expanding the exploration range of the solution space.

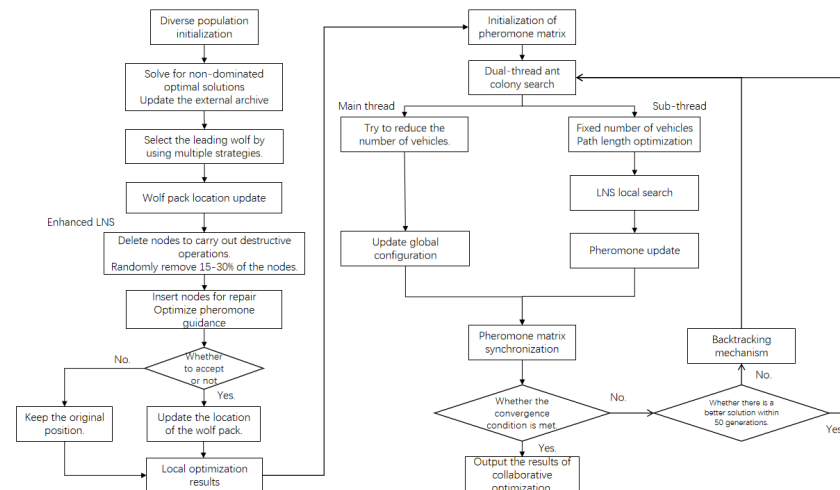


Figure 1. Overall Flowchart of the Algorithm.

After the LNS optimization, the algorithm enters the dual-thread ant colony collaboration stage. The main thread, ACS_Time, focuses on reducing the travel cost by dynamically adjusting the pheromone concentration and local search strategies to continuously optimize the path. The sub-thread, ACS_Vehicle, aims to minimize the number of vehicles used by intelligently adjusting the customer allocation plan to lower travel costs. The two threads achieve collaborative optimization through a shared global pheromone matrix, where the pheromone of high-quality paths is enhanced, while that of low-quality paths gradually evaporates. This collaborative mechanism enables the two optimization goals to progress simultaneously, preventing a single goal from overly dominating the search direction. The dual-thread system inherits the high-quality solution set from the large neighborhood optimization and continuously improves the solution quality through an elite solution retention mechanism. To address the issue of imbalance in the optimization progress of the dual threads, the algorithm introduces a backtracking mechanism. When the number of vehicles decreases too rapidly without sufficient optimization of the travel cost, it automatically reverts to the previous vehicle number stage for researching, ensuring the convergence of multi-objective collaboration.

The algorithm uses both the number of iterations and the quality of the solutions as termination conditions. By default, 500 iterations are set, and the rate of change in the objective function is monitored. When no improvement occurs after 200 consecutive iterations, the algorithm is determined to have converged. Finally, the algorithm outputs the optimal compromise solution from the Pareto solution set and provides a convergence curve for analyzing the optimization process.

The innovation of the algorithm lies in achieving a complete process from rapid initial solution generation to refined optimization through multi-stage collaboration; synchronously advancing multiple optimization goals through a dual-thread ant colony system; and intelligently balancing different optimization goals through a dynamic pheromone update mechanism. Experimental results show that compared with traditional single-

objective optimization methods, this algorithm has significant improvements in vehicle utilization efficiency and algorithm running speed, providing an effective solution for complex logistics distribution problems.

4.2. MOGWO Initialization

4.2.1. Initial Solution Generation Strategy

This paper adopts an improved nearest neighbor heuristic algorithm to generate the initial solution. The algorithm flow is shown in Algorithm 1. The algorithm introduces four improvements on the basis of the traditional nearest neighbor method, namely demand priority sorting, dynamic capacity constraints, real-time time window verification, and fault-tolerant mechanism. Firstly, the customer points are sorted in descending order of demand. This dual sorting strategy not only considers the required load volume but also takes into account the opening time of the time window, ensuring that high-demand customer points are served first. During the path construction process, an elastic threshold control is introduced. When the vehicle loading volume reaches 80% of the capacity limit, the return-to-depot strategy is immediately triggered, which not only avoids the risk of overloading but also improves the vehicle utilization rate. At the same time, the time window constraints are strictly verified each time a customer point is assigned, and customer points who cannot meet the time requirements are automatically assigned to subsequent vehicles. And a multi-level fault-tolerant mechanism is used to ensure the feasibility of the path: first, the basic path repair method is used to force all paths to start and end at the depot node (0). When it is detected that the amount of delivered goods plus the goods required by the next node is greater than the vehicle's cargo capacity, the depot node is immediately inserted to split the path. This fourfold guarantee mechanism significantly improves the feasibility and quality of the initial solution and provides a high-quality initial population for the subsequent MOGWO optimization.

Algorithm 1 The nearest neighbor heuristic algorithm generates the initial solution.

```

1: vehicle_capacity = graph.vehicle_capacity
2: rho = 0.15
3: path, total_distance, vehicle_count
4: path = [0]
5: current_load = 0
6: current_time = 0
7: while there are unvisited nodes do
8:   if current_load  $\geq 0.8 \times$  vehicle_capacity then
9:     distance_to_depot = graph.node_dist_mat[current_node][0]
10:    path.append(0)
11:    current_load = 0
12:    current_time = 0
13:   else
14:     candidates = [n for n in unvisited nodes if current_load + n.demand  $\leq$  vehicle_capacity and current_time + dist  $\leq$  n.due_time]
15:     if no candidate nodes then
16:       path.append(0)
17:       current_load = 0
18:       current_time = 0
19:     else
20:       next_node = arg mingraph.node_dist_mat[current_node][candidates]
21:       update path, current_load, current_time
22:     end if
23:   end if
24: end while
25: return path, total_distance, vehicle_count

```

4.2.2. Grey Wolf Population Optimization Mechanism

The specific process of the multi-objective grey wolf optimization mechanism is shown in Algorithm 2. The algorithm adopts a dynamic selection strategy based on the Pareto dominance relationship: first, the population is divided into different levels through fast non-dominated sorting, and then the leader wolves are selected from the non-dominated solution set of the highest level according to specific optimization objectives. Specifically: the α wolf corresponds to the solution with the least number of vehicles, reflecting the pursuit of minimizing operational costs; the β wolf corresponds to the solution with the shortest path distance, dedicated to optimizing travel efficiency; the δ wolf selects the solution with the highest congestion degree, maintaining population diversity by calculating the neighborhood density in the objective space. During the search process, the leader wolves are updated using an elite retention strategy. After each generation of optimization, the algorithm combines the current population with the external archive, re-executes the non-dominated sorting, and updates the three-level leader wolves. This mechanism ensures the inheritance of high-quality solutions and effectively avoids premature convergence through the diversity maintenance function of the δ wolf. In terms of encoding design, an intuitive integer sequence encoding method is adopted, where each solution is represented as an ordered sequence containing depot nodes (0) and customer point numbers. For example, the encoding [0, 3, 5, 0, 2, 4, 0] represents two travel costs: the first vehicle's path is depot $\rightarrow$ customer point 3 $\rightarrow$ customer point 5 $\rightarrow$ depot; the second vehicle's path is depot $\rightarrow$ customer point 2 $\rightarrow$ customer point 4 $\rightarrow$ depot. This encoding method fully retains the topological structure and sequence relationship of the path, facilitating subsequent path optimization.

The design of the fitness function adopts a hybrid evaluation mechanism based on Pareto rank. For the selection of leader wolves, independent evaluations are conducted for the three optimization objectives: the fitness of the α wolf mainly considers the number of vehicles; the β wolf focuses on the path distance; and the δ wolf emphasizes the calculation of congestion degree. For ordinary individuals, a weighted sum method is used to convert multi-objectives into a single objective for comprehensive evaluation, where the weight coefficients are dynamically adjusted with the number of iterations. This dynamic adjustment strategy is reflected in: in the early stage of optimization, slight constraint violations are allowed to expand the search space, while in the later stage, the constraint conditions are strengthened to guide the population towards the feasible region. Specifically, the capacity constraint penalty is achieved by accumulating the overload amount: when the vehicle load exceeds the rated capacity, the penalty value is equal to the weighted sum of the overloaded part; the time window penalty quantifies the deviation between the service time and the required time window of the customer point.

As shown in Figure 2, the algorithm realizes the collaborative update of the population position through the leadership wolf guidance mechanism. In each generation of optimization, the α , β , and δ three leading wolves respectively represent the elite solutions in different optimization directions. Ordinary wolves accept the guidance of the leading wolves with a 70% probability and update their positions through sequential crossover operations. The position update strategy reflects the intelligent characteristics of the grey wolf group's collaborative hunting: in the encirclement stage, the three leading wolves respectively guide the search direction from the dimensions of the number of vehicles, travel costs, and population diversity; in the attack stage, new solutions are generated through crossover operations and only feasible solutions are strictly retained. This mechanism not only ensures population diversity but also enables continuous convergence toward high-quality solution regions. The iteration is terminated after 50 consecutive generations, effectively balancing the solution efficiency and quality.

Algorithm 2 Multi-objective Grey Wolf Optimization Mechanism.**Require:** graph: VrptwGraph

1: num_wolves = 30

2: max_iter = 50

3: archive_size = 20

4: $\sigma = 0.6, \phi = 0.4$ 5: $\mu_{init} = 0.1$ **Ensure:** Pareto_front: List[(vehicle_num, total_distance)]

6: Non-dominated solution set:

7: Wolf pack position: Randomly generate feasible solutions

8: Fitness evaluation: Calculate two objective values for each solution

9: Non-dominated solution set α Wolf, β Wolf, δ Wolf

10: Pareto frontier archive = []

11: Fitness assessment:

12: Calculate the multi-objective fitness for each solution:

13: $fitness = \sigma \times vehicle_num + \phi \times total_distance + \mu \times penalty$ 14: $penalty = Volume\ violation\ amount + Quantity\ of\ time\ window\ violations$ 15: $\mu = \mu \times (1 + iter / max_iter)$ # Dynamic penalty coefficient16: **while** iter $\leq$ max_iter **do**

17: a. To each wolf:

18: i. According to the location information of α, β, δ update its own position

19: ii. When generating new solutions, apply the path repair strategy.

20: b. Calculate the Pareto frontier of all solutions:

21: Use non-dominated sorting to filter the solutions in the current population and the archive.

22: Calculate the congestion degree of the solution (to ensure the diversity of the solution)

23: Update the archive to the top archive_size optimal solutions.

24: c. Update α Wolf, β Wolf, δ Wolf25: α Wolf = The solution with the smallest number of vehicles in the archive.26: β Wolf = The solution with the smallest total distance in the archive.27: δ Wolf = The solution with the highest congestion in the archive.

28: d. Dynamically adjust parameters:

29: The convergence factor a linearly decreases from 2 to 0.

30: Random vector $r1, r2 \in [0, 1]$ 31: **end while**32: **return** archive # Output the Pareto optimal solution set

As shown in Figure 3, the solution is sequentially crossed in the form of path recombination to simulate the cooperative search behavior of wolf packs. The crossover process first randomly selects a path split point, which is usually located at the customer node rather than the warehouse node, to ensure the integrity and feasibility of the path segments. Specifically, the selection of the split point must meet two conditions: one is to avoid disrupting the continuity of the single vehicle path, and the other is to ensure that each sub-path after segmentation still meets the vehicle capacity constraints. The segment inheritance operation achieves knowledge transfer by integrating high-quality segments from different paths: the ordinary wolf C follows the leader wolf A [0, 1, 2, 0, 3, 4, 0] that seeks the minimum number of vehicles and the leader wolf B [0, 5, 6, 0, 7, 8, 0] that seeks the lowest cost for hunting, and inherits the paths of the two leader wolves. It combines the front segment path [0, 1, 2] of leader wolf A with the back segment path [0, 7, 8, 0] of the other leader wolf B to form a new hunting path [0, 1, 2, 0, 7, 8, 0]. Since direct combination may lead to repeated or missing visits to customer points, conflict detection and resolution must be carried out subsequently: repeated customer numbers are identified through the node mapping table, and conflicts are eliminated using the nearest neighbor replacement strategy, ultimately generating the optimized path [0, 1, 2, 5, 0, 7, 8, 6, 0]. This crossover

strategy not only maintains the legitimacy of the solution structure but also effectively improves the search efficiency through the inheritance of elite path segments.

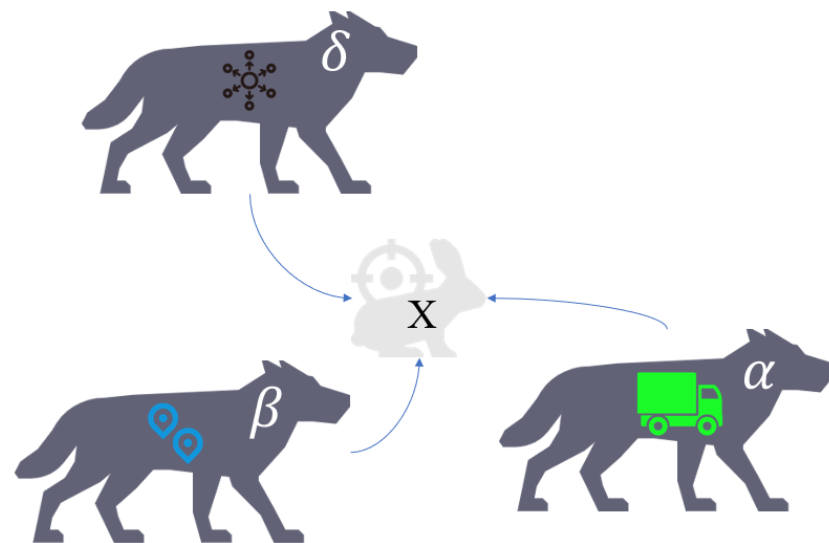


Figure 2. Schematic Diagram of the Directions of the Three Leading Wolves: Alpha, Beta, and Delta.

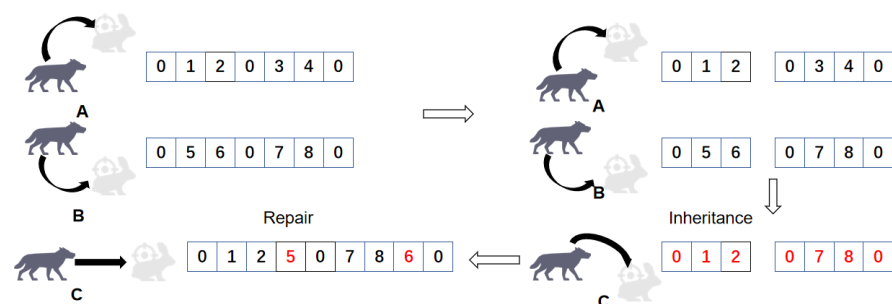


Figure 3. Simulates the path crossing of wolf pack cooperative search behavior.

As shown in Figure 4, the mutation operation is carried out in the form of solution reset, and a complete reconstruction strategy is adopted to simulate the exploration behavior of the wolf pack, thereby maintaining the diversity of the population. When a wolf individual does not accept the guidance of the leading wolf (with a trigger probability of 30%), the system initiates an intelligent reconstruction mechanism to generate a new path. The reconstruction process is divided into three stages: Firstly, the hunting is completely rearranged to [1, 2, 3, 4], breaking the original hunting path [0, 1, 2, 0, 3, 4, 0]; then, it is sorted in descending order of urgency [4, 2, 1, 3] to ensure that the places with higher urgency are reached first; finally, the path is segmented based on the vehicle capacity as a dynamic threshold. When the cumulative demand exceeds the capacity constraint, the warehouse node (number 0) is immediately inserted to complete the current hunting path and initialize a new hunting path [0, 4, 2, 0, 1, 3, 0]. After the reconstruction is completed, the system will verify that the new path meets all the constraints, including time window constraints, capacity constraints, and vehicle number constraints. This mutation strategy based on complete reconstruction breaks through the small perturbation limitations of traditional single-point or exchange mutations. By large-scale solution reconstruction, it effectively avoids the algorithm getting trapped in local optima and significantly increases the probability of obtaining the global optimal solution.

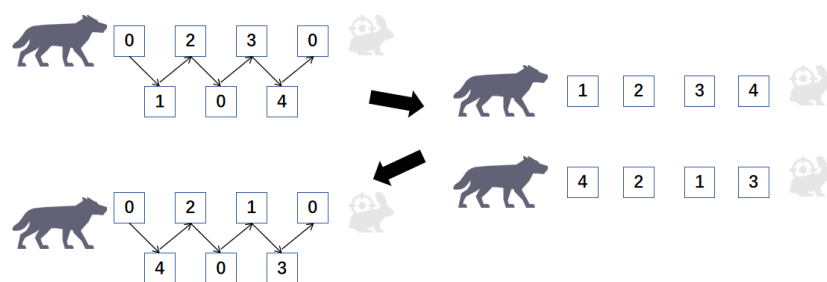


Figure 4. Mutation of the exploration behavior of the simulated wolf pack.

The algorithm adopts an elite retention strategy to ensure that high-quality solutions are not lost. In each generation of optimization, all individuals are evaluated for fitness. When a solution better than the current optimal one is found, the global optimal record is immediately updated. The selection mechanism has a dual guiding role: on the one hand, it enhances the local development ability through the guidance of the leading wolf, quickly converging to the current optimal region; on the other hand, it maintains the global exploration ability through intelligent mutation, continuously exploring new solution spaces.

4.3. The Iterative Process of the Dual-Threaded ACO

4.3.1. Thread Division of Labor and Coordination Mechanism

The algorithm adopts a parallel computing architecture, with two independent threads, ACS_Time and ACS_Vehicle, respectively optimizing the number of vehicles and travel costs. The ACS_Time thread is used for optimizing the travel cost, seeking the shortest travel cost under the premise of a fixed number of vehicles; the ACS_Vehicle thread is used to reduce the number of vehicles used, achieving cost savings by reducing the number of service vehicles. The two threads achieve information exchange through sharing the pheromone matrix and the global path queue, forming a co-evolution mechanism. When either thread discovers an improved solution, it immediately notifies the other through the path queue to update the search direction. This design enables the algorithm to simultaneously advance towards the two goals of reducing the number of vehicles and shortening the travel cost.

Specifically, the ACS_Time thread focuses on path optimization using an elite strategy, and its state transition rule introduces a dynamic weight adjustment mechanism: in the early stage of the algorithm (iteration count $t < 10$), $\sigma = 1.5$ and $\phi = 2.0$ are set to enhance exploration ability, and later ($t \geq 10$), they are adjusted to $\sigma = 2.0$ and $\phi = 1.5$ to strengthen pheromone guidance. Meanwhile, the ACS_Vehicle thread adopts an adaptive evaporation strategy, and its evaporation rate ρ is determined by the formula:

$$\rho = \rho_b + 0.1 \times (V_{\max} - V_{\text{current}}) \quad (13)$$

Here ρ_b is the initial evaporation rate, $V_{\max}$ is the initial number of vehicles, and V_{current} is the current number of vehicles. When one vehicle is successfully reduced, the depth optimization mode of 2-opt local search is triggered, and the path segments are optimized for three rounds of iterations.

The local search adopts a hybrid strategy combining path segment exchange and insertion optimization. In the path segment exchange stage: Select segments of no more than 5 lengths from two paths for cross-exchange, verify the feasibility, and update the path. In the insertion optimization stage: Conduct feasibility checks on unvisited customer points and find the optimal insertion position. The innovation lies in the dynamic adjustment of search intensity: When there is no improvement for 10 consecutive generations, increase the exchange segment length from 5 to 8 to expand the search range.

4.3.2. Cooperative Update of Pheromone Matrix

The pheromone update system adopts a two-layer collaborative architecture: the base layer performs standard global updates, where the ρ value is set differently based on the thread type (ACS_Time is fixed at 0.1, and ACS_Vehicle is dynamically adjusted); the collaborative layer achieves cross-thread pheromone fusion through a shared memory queue, synchronizing the elite solution set every five iterations. The update formula is extended to:

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

Here $\Delta\tau_{ij}^k = \frac{Z}{D_k + \varepsilon}$, $\varepsilon = 0.01$ is a smoothing factor, m is the total number of ants participating in the information update in the current iteration, and Z is the quality coefficient ($Z = 2.0$ for elite solutions and $Z = 1.0$ for ordinary solutions). This design enables the algorithm to maintain convergence while enhancing the diversity of solutions.

4.3.3. Backtracking Mechanism

Due to the pursuit of balance in the target through the dual-thread method, the ACS_Time thread seeks the shortest driving path under the condition of a fixed number of vehicles, while the ACS_Vehicle thread focuses on reducing the number of vehicles used. Specifically, when the number of vehicles in the ACS_Time thread is K , the number of vehicles in the ACS_Vehicle thread is $K - 1$. This may lead to a situation where the ACS_Vehicle thread discovers a path with fewer vehicles and lower cost before the ACS_Time thread finds the shortest path for the current number of vehicles. This situation occurs repeatedly, causing the number of vehicles to reach the minimum but the cost not to reach the lowest, thus disrupting the balance of multi-objective optimization. To reduce the occurrence of this situation, a backtracking mechanism is introduced into the ant colony algorithm. When no better solutions are found for both K vehicles and $K - 1$ vehicles within 50 generations, the number of vehicles in the ACS_Time and ACS_Vehicle threads are respectively backtracked to $K + 1$ and K , and another 50 generations are iterated. This process is repeated until no better solutions are found for 200 consecutive generations, at which point the result is considered to have converged and the current result is output. To verify the effectiveness of the backtracking mechanism, ablation experiments were conducted in the experimental section.

4.4. Co-Optimization of LNS and MOGWO-DTACO Algorithm

In the grey wolf optimization stage, the algorithm significantly enhances the quality and diversity of the Pareto solution set through an embedded adaptive LNS. This mechanism adopts a triple optimization design: Firstly, it implements a dynamic destruction–repair framework based on customer point clustering analysis, intelligently removing 15% to 30% of associated customer points while maintaining the structural integrity of path fragments. Secondly, it builds a multi-dimensional reconstruction evaluation criterion that integrates path continuity, distance cost, and time window urgency. At the same time, it establishes an elite solution retention mechanism, only accepting solutions with improvements in the comprehensive objective function and updating the leader wolf's position in real time. The optimization process integrates a three-layer structure of node exchange, intelligent insertion, and path fragment optimization, specifically including: using the node exchange operator to break the local optimum limitation, prioritizing the scheduling of key customer points based on time window urgency, and eliminating path intersections to enhance transportation efficiency. The adaptive activation mechanism performs periodic deep optimization (triggered every 5 generations), automatically adjusting the search intensity based on the improvement in the objective function, with the

destruction rate linearly decreasing from the initial value of 0.3 to 0.1, effectively balancing exploration and exploitation.

Under the framework of the dual-thread ant colony system, LNS, as the core optimization module, achieves symmetrical optimization for multiple objectives. The main thread focuses on minimizing path distance, employing a time-sensitive node removal strategy to eliminate path intersections and compress travel costs; the sub-thread, on the other hand, aims to minimize the number of vehicles, optimizing customer point allocation through load balancing path reorganization techniques to enhance vehicle utilization. The dual threads adopt an intelligent response mechanism: the main thread automatically triggers optimization when there is no improvement in travel cost for two consecutive generations, while the sub-thread initiates deep reorganization when the number of vehicles does not decrease. Both threads achieve strategy synergy through a shared pheromone matrix, with the main thread implementing time window-sensitive optimization (prioritizing the processing of urgent customer points and eliminating conflict areas), and the sub-thread performing load rebalancing operations (identifying inefficient paths and reorganizing customer point resource allocation), forming an optimization architecture that decouples objectives and complements capabilities.

5. Analysis of Experimental Results

5.1. Experimental Design

This experiment selects the publicly available Solomon dataset, which contains 56 cases. Each case consists of one depot and 100 customer points, and these customers are distributed within a 100×100 Cartesian coordinate system. According to the distribution characteristics of customer points, the Solomon dataset is divided into three types: clustered type (C type), random type (R type), and mixed type (RC type). Among them, the characteristics of type 1 are narrow time windows and small vehicle load capacity, while type 2 has wider time windows and larger vehicle load capacity. This paper conducts experiments on 25 customer points, 50 customer points, and 100 customer points of all datasets respectively.

The algorithm parameters are set as follows: the gray wolf population size is 30, the number of iterations of the multi-objective gray wolf optimization algorithm is 50 generations, the ant population size is 30, the heuristic information weight is 5, the initial exploration probability is 0.15, the pheromone evaporation coefficient is 0.3, and the maximum number of iterations of the ant colony algorithm is 500 generations.

5.2. Performance Verification of the MOGWO-DTACO Algorithm Improvement

To verify the effectiveness of the overall algorithm, this paper designs systematic comparative experiments and ablation experiments. Firstly, by comparing the improved MOGWO-DTACO algorithm with the original DTACO algorithm on multiple standard datasets, the performance of the algorithm's initialization strategy and collaborative optimization mechanism is evaluated respectively. The results confirm the significant advantages of the improved algorithm in terms of convergence speed and solution quality. Secondly, to determine the reasonable values of key parameters, this paper further designs parameter sensitivity experiments. Through multiple configuration tests of parameters such as the pheromone evaporation factor, the optimal parameter combination is determined. Finally, to clarify the independent contribution of the backtracking mechanism in balancing the number of vehicles and travel costs, ablation experiments are conducted on the R1-100 dataset. The results show that after introducing the backtracking mechanism, the algorithm can obtain solutions with fewer vehicles or lower total costs in most cases, effectively alleviating the phenomenon of premature convergence of the number of vehicles while the

travel cost is not fully optimized. Through multiple sets of experiments, the effectiveness of the algorithm in solving the VRPTW problem is demonstrated.

To verify the superiority of the MOGWO-DTACO algorithm proposed in this paper, we conducted comparisons with single-objective algorithms and the multi-objective algorithm MOGWO-PSO under the same experimental conditions. As shown in Table 1, the experimental results on 100 customer points in the C-class dataset indicate that: Firstly, the multi-objective optimization algorithms MOGWO-PSO and MOGWO-DTACO generally outperform the single-objective algorithms GWO and ACO in terms of solution quality. Secondly, among the multi-objective algorithms, the MOGWO-DTACO algorithm proposed in this paper demonstrates a faster convergence speed, with an average of 311 iterations required to reach a similar or better solution, which is significantly less than the 3390 iterations of the MOGWO-PSO algorithm. This proves the effectiveness of the improved strategies introduced in this paper.

Table 1. Comparison of Optimization Results on C-Series Benchmark Problems (100 Customers).

Case Name	GWO	ACO	MOGWO-PSO	MOGWO-DTACO
C101-100	828.94-10	828.94-10	828.94-10	828.94-10
C102-100	828.94-10	828.94-10	828.94-10	828.94-10
C103-100	1124.66-10	1083.90-10	828.06-10	828.06-10
C104-100	1109.15-10	1092.26-10	824.78-10	824.78-10
C105-100	828.94-10	828.94-10	828.94-10	828.94-10
C106-100	828.94-10	828.94-10	828.94-10	828.94-10
C107-100	828.94-10	828.94-10	828.94-10	828.94-10
C108-100	828.94-10	828.94-10	828.94-10	828.94-10
C109-100	828.94-10	828.94-10	828.94-10	828.94-10
C201-100	591.56-3	591.56-3	591.56-3	591.56-3
C202-100	695.56-3	865.14-3	591.56-3	591.56-3
C203-100	768.18-3	882.70-3	591.17-3	591.17-3
C204-100	837.71-3	942.05-3	590.60-3	590.60-3
C205-100	588.88-3	588.88-3	588.88-3	588.88-3
C206-100	588.49-3	588.49-3	588.49-3	588.49-3
C207-100	591.56-3	591.56-3	588.29-3	588.29-3
C208-100	599.20-3	599.20-3	588.32-3	588.32-3

5.2.1. Comparison Between DTACO and MOGWO-DTACO

Compared with the original DTACO algorithm, this paper makes improvements to the initialization and collaboration mechanism of the algorithm. Therefore, to verify the superiority of the improvements, the first 50 generations and the final results of the algorithm are compared respectively. 18 datasets are selected from the C, R and RC type datasets, and tests are conducted on different numbers of customer points. Each dataset is tested 10 times with different numbers of customer points. The results are shown in Table 2.

The experimental comparisons in Table 2 show that at the 50th generation, the improved MOGWO-DTACO algorithm generally outperforms the DTACO algorithm, demonstrating the superiority of the initialization improvement. For some datasets such as C107-100 and R101-25, the optimal solution has already been obtained after the initialization of the MOGWO algorithm at the 50th generation. Through the comparison of the final results, it can also prove the effectiveness of the improved collaborative optimization strategy in this paper. By counting the number of iterations, the average convergence generation of the MOGWO-DTACO algorithm is approximately 311 generations. Compared with the DTACO algorithm up to 500 generations, the average number of iterations is reduced by 189 generations, which proves the improvement in algorithm speed.

Table 2. Comparison Results of DTACO and MOGWO-DTACO.

Case Name	DTACO(50)	MOGWO-DTACO(50)	DTACO(end)	MOGWO-DTACO(end)
C101-25	258.92-3	217.3-3	191.8-3	191.8-3
C104-50	456.29-10	359.5-7	358.9-5	358.9-5
C107-100	993.41-10	828.94-10	828.94-10	828.94-10
C202-25	273.7-1	273.7-1	215.5-2	215.5-2
C205-50	532.47-9	404.29-9	361.4-3	361.4-3
C208-100	827.4-5	788.1-6	588.32-3	588.32-3
R101-25	645.19-14	618.3-8	629.1-8	618.3-8
R104-50	713.8-14	695.79-12	661.87-7	631.6-6
R107-100	1312.0-22	1180.5-21	1272.31-11	1272.31-11
R202-25	779.89-7	731.7-7	457.6-4	411.5-4
R205-50	831.0-8	742.0-6	702.6-4	702.6-4
R208-100	832.2-6	755.0-5	809.8-5	721.04-4
RC101-25	574.7-10	529.8-8	462.2-4	462.2-4
RC104-50	610.0-9	600.36-8	549.0-6	546.5-5
RC107-100	1548.1-20	1421.9-18	1485.36-11	1485.36-11
RC202-25	402.41-10	338.8-5	340.23-5	338.8-3
RC205-50	992.6-8	633.9-7	632.0-5	632.0-5
RC208-100	1005.9-7	918.62-5	875.37-5	875.37-5

5.2.2. Comparison of Different Parameters of MOGWO-DTACO

To explore the influence of specific parameters on the MOGWO-DTACO algorithm, this paper designs a comparative experiment to verify the rationality of the parameters selected in this paper. The specific experiments are shown in Table 3. Here, 100 customer points from all C-type datasets are selected as the experimental dataset. The evaporation probability of pheromone is taken as the experimental object, and $\rho = 0.2, 0.3, 0.4$ are respectively selected as the test parameters.

Table 3. Comparison Results of Different Parameters of MOGWO-DTACO.

Case Name	Rho = 0.2	Rho = 0.3	Rho = 0.4
c101-100	828.94-10	828.94-10	828.94-10
c102-100	828.94-10	828.94-10	828.94-10
c103-100	1128.7-10	828.06-10	1083.9-10
c104-100	1202.22-10	824.78-10	1092.26-10
c105-100	830.46-10	828.94-10	828.94-10
c106-100	879.71-10	828.94-10	828.94-10
c107-100	856.29-10	828.94-10	828.94-10
c108-100	997.63-10	828.94-10	938.51-10
c109-100	1200-10	828.94-10	969.44-10
c201-100	591.56-3	591.56-3	591.56-3
c202-100	947.31-3	591.56-3	981.21-3
c203-100	965.6-3	591.17-3	981.21-3
c204-100	1063.16-3	590.60-3	986.71-3
c205-100	893.88-3	588.88-3	648.81-3
c206-100	853.53-3	588.49-3	716.28-3
c207-100	1083-3	588.29-3	727.05-3
c208-100	893.88-3	588.32-3	719.92-3

The data results in Table 3 can verify the validity of the parameters selected in this paper. Except for the pheromone evaporation probability, the other parameters have also been tested through the same experiments to determine the optimal parameter settings.

5.2.3. Validation of the Effectiveness of the Backtracking Mechanism

To verify the effectiveness of the backtracking mechanism designed in this paper, an ablation experiment was conducted on the R1-100 dataset. Each dataset was run ten times, and the best solution was recorded. The results were compared with those of the Scpso algorithm, as shown in Table 4.

As shown in Table 4, the results with the backtracking mechanism introduced are far better than those without it. Moreover, in the R102-100 dataset, the results are better than those of the ACS-BSO algorithm. There are also cases like in the R103-100 algorithm where the number of vehicles is less than that of the ACS-BSO algorithm, but the cost slightly increases. In the R107-100 dataset, due to the increase in the number of vehicles caused by the backtracking mechanism, some solutions still remain at a relatively high cost and number of vehicles when the algorithm reaches the set maximum number of iterations and is forced to converge. This is also a direction that needs improvement in future research.

Table 4. Ablation Experiment of Backtracking Mechanism.

Case Name	ACS-BSO [27]	No Backtracking Mechanism	Backtracking Mechanism
R101-100	1642.88-20	1659.34-19	1659.34-19
R102-100	1479.55-18	1559.88-18	1474.75-18
R103-100	1225.31-15	1450.89-16	1240.97-14
R104-100	1002.62-12	1012.97-10	1012.97-10
R105-100	1365.66-15	1540.9-15	1416.7-15
R106-100	1249.51-13	1390.56-11	1249.51-13
R107-100	1091.21-11	1272.31-11	1191.24-14
R108-100	960.23-10	960.23-10	960.23-10

5.3. Experimental Results

To verify the effectiveness of the algorithm, experiments were conducted in PyCharm Community Edition 2024.1. The standard Solomon dataset was selected as the experimental case for training. Each dataset was run 10 times. For datasets with 25 and 50 customer points, the results were retained to one decimal place, while for the dataset with 100 customer points, the results were retained to two decimal places. For example, C101-100 indicates that the C101 dataset with a total of 100 customer points was tested. The operation results are shown in Tables 5 and 6 respectively.

Based on the data in Table 5, it can be concluded that MOGWO-DTACO yields results mostly consistent with BKSs when there are 25 and 50 customer points. In datasets such as R102 and R103, the optimal solution is better than BKSs. When there are 25 customer points, the average lowest cost slightly increases by 0.54% and the average number of vehicles decreases by 1.55%. When there are 50 customer points, the average lowest cost is almost the same and the average number of vehicles decreases by 3.53%. Since the algorithm in this paper adopts multi-objective optimization, it optimizes the number of vehicles and cost separately. Therefore, in some datasets, such as R210-25, although the lowest cost is not as low as BKSs, the number of vehicles is reduced by one. Under the condition of a small cost gap, reducing the number of vehicles makes such solutions acceptable. The results show that the proposed algorithm outperformed the BKS results on eight datasets, while on six datasets, it achieved a reduction in the number of vehicles at the cost of a slight increase in travel cost.

Table 5. Results of all datasets when the customer points are 25 and 50.

Dataset Number	25 Customer Points		50 Customer Points	
	BKS	MOGWO-DTACO	BKS	MOGWO-DTACO
C101	191.3-3	191.8-3	362.4-5	363.2-5
C102	190.3-3	190.7-3	361.4-5	362.2-5
C103	190.3-3	190.7-3	361.4-4	362.2-5
C104	186.9-3	187.4-3	359.0-5	358.9-5
C105	191.3-3	191.8-3	362.4-5	363.2-5
C106	191.3-3	191.8-3	362.4-5	363.2-5
C107	191.3-3	191.8-3	362.4-5	363.2-5
C108	191.3-3	191.8-3	362.4-5	363.2-5
C109	191.3-3	191.8-3	362.4-5	363.2-5
C201	214.7-2	215.5-2	360.2-3	361.8-3
C202	214.7-2	215.5-2	360.2-3	361.8-3
C203	214.7-2	215.5-2	359.8-3	361.4-3
C204	213.1-1	213.9-1	353.4-2	351.7-2
C205	214.7-2	215.5-2	359.8-3	361.4-3
C206	214.7-2	215.5-2	359.8-3	361.4-3
C207	214.5-2	215.3-2	359.6-3	361.2-3
C208	215.4-2	215.4-2	350.5-2	352.1-2
R101	617.1-8	618.3-8	1047.0-13	1047.0-13
R102	547.1-7	548.1-7	944.9-12	932.7-11
R103	454.6-5	455.7-5	722.9-9	784.6-8
R104	416.9-4	418.0-4	631.2-6	631.6-6
R105	530.5-6	531.5-6	906.6-10	923.4-10
R106	465.4-5	466.5-5	793.6-8	795.3-8
R107	424.3-4	425.3-4	720.4-7	720.4-7
R108	397.3-4	398.3-4	618.2-6	630.3-6
R109	441.3-5	442.6-5	803.2-8	796.9-8
R110	444.1-4	445.2-5	724.9-8	717.9-8
R111	428.8-4	429.7-4	724.9-8	724.9-8
R112	393.0-4	394.1-4	651.1-6	643.7-6
R201	463.3-4	464.4-4	800.7-6	801.8-5
R202	410.5-4	411.5-4	712.2-5	774.6-3
R203	391.4-3	392.3-3	606.4-5	639.1-4
R204	355.0-2	355.9-2	509.5-2	511.1-2
R205	393.0-3	394.1-3	703.3-5	702.6-4
R206	374.4-3	376.0-3	647.0-5	647.0-5
R207	361.6-3	362.6-3	584.6-4	617.7-2
R208	328.2-1	329.3-1	487.7-2	491.0-2
R209	370.7-2	371.6-2	600.6-4	608.1-4
R210	404.6-3	410.60-2	663.4-5	647.9-4
R211	350.9-2	351.9-2	551.3-3	544.9-3
RC101	461.1-4	462.2-4	957.9-9	957.9-9
RC102	351.8-3	352.7-3	844.3-8	824.0-7
RC103	332.8-3	333.9-3	712.6-6	712.6-6
RC104	306.6-3	307.1-3	546.5-5	546.5-5
RC105	411.3-4	412.4-4	888.9-9	857.0-8
RC106	345.5-3	346.5-3	791.9-7	724.7-6
RC107	298.3-3	298.9-3	664.5-6	645.6-6
RC108	294.5-3	295.0-3	598.1-6	599.2-6
RC201	360.2-3	361.2-3	684.8-5	687.4-5
RC202	338.0-3	338.8-3	613.6-5	615.1-5
RC203	326.9-3	356.2-2	555.3-4	556.5-4
RC204	299.7-3	300.2-3	444.2-3	444.9-3
RC205	338.0-3	338.9-3	631.0-5	632.0-5
RC206	324.0-3	344.9-2	610.0-5	611.7-5
RC207	298.3-3	298.9-3	558.6-4	559.9-4
RC208	269.1-2	269.6-2	535.8-2	535.8-2
Avg	333.5-3.23	335.3-3.18	587.7-5.39	589.0-5.20

Table 6. Results of Some Data Sets When the Customer Point is 100.

Case Name	BKS	HHSA [28]	SCPSO [29]	MOGWO-DTACO
C101-100	828.94-10	828.94-10	828.94-10	828.94-10
C102-100	828.94-10	828.94-10	828.94-10	828.94-10
C103-100	828.06-10	828.06-10	828.06-10	828.06-10
C104-100	824.78-10	824.78-10	846.78-10	824.78-10
C105-100	828.94-10	828.94-10	828.94-10	828.94-10
C106-100	828.94-10	828.94-10	828.94-10	828.94-10
C107-100	828.94-10	828.94-10	828.94-10	828.94-10
c108-100	828.94-10	828.94-10	828.94-10	828.94-10
c109-100	828.94-10	828.94-10	854.78-10	828.94-10
C201-100	591.56-3	591.56-3	591.56-3	591.56-3
C202-100	591.56-3	591.56-3	591.56-3	591.56-3
C203-100	591.17-3	591.17-3	591.17-3	591.17-3
C204-100	590.60-3	596.55-3	596.55-3	590.60-3
C205-100	588.88-3	588.88-3	588.88-3	588.88-3
C206-100	588.49-3	588.49-3	588.49-3	588.49-3
C207-100	588.29-3	588.29-3	588.29-3	588.29-3
C208-100	588.32-3	588.32-3	588.32-3	588.32-3
R101-100	1642.88-20	1656.5-19	1655.87-20	1659.34-19
R108-100	947.55-10	984.81-10	978.49-11	960.23-10
R203-100	874.87-6	882.29-6	882.44-5	1043.74-3
R208-100	705.45-4	718.54-4	710.3-4	721.04-4
RC103-100	1261.67-11	1290.3-12	1305.4-12	1305.4-12
RC105-100	1518.58-15	1564.8-15	1553.58-15	1689.75-14
RC201-100	1265.56-9	1283.82-7	1286.62-7	1489.21-6
RC207-100	966.08-6	984.93-6	966.08-6	1146.55-3

As shown in Figure 5, the route maps with the lowest cost and the least number of vehicles, as well as the number of iterations, are presented from left to right as: C204-50, R109-50, and RC102-50. Visually, it can be seen that this algorithm maintains extremely high operational efficiency while having the ability to find the optimal solution. For 25 customer points, the average number of iterations is approximately 216 generations; for 50 customer points, it is about 264 generations. Specifically: for the C-type dataset, the multi-objective grey wolf optimization algorithm quickly generates high-quality initial solutions by using path crossover operations and elite retention strategies, and the solution quality approaches the optimal within 50 generations. Secondly, for the R-type dataset, the algorithm adopts diversity initialization and adaptive exploration mechanisms, maintaining population diversity through the calculation of the crowding degree of δ wolves, effectively balancing the minimization of the number of vehicles and the optimization of path costs. Additionally, for the RC-type dataset, the algorithm achieves unified optimization of clustered and random areas through dynamic strategies and the division of labor and collaboration among leader wolves. Figure 6 shows the path planning of C101-100 and C208-100 in the C-type dataset. From the spatially more clustered C101 to the more evenly distributed customer points in C208, the algorithm can stably find the optimal path, further verifying the good stability of the algorithm in the dataset.

Table 6 shows that the MOGWO-DTACO algorithm found the best-known solution (BKS) in the C dataset when there were 100 customer points, while its performance in the R and RC datasets was comparable to that of the HHSA and SCPSO algorithms. The performance degradation of the algorithm on R and RC instances mainly stems from the insufficiency of the current algorithm framework in adapting to the characteristics of random and mixed distributions. Firstly, the population initialization mechanism overly relies on the clustering assumption, making it difficult to generate high-quality initial

solutions in R instances where customer points are randomly distributed, which leads the subsequent optimization process to get trapped in local optima. Secondly, the leader wolf collaboration mechanism shows limitations in a dispersed solution space, resulting in an imbalance between the goals of minimizing the number of vehicles and optimizing the path cost, which restricts the uniformity of the Pareto solution set distribution. Moreover, the fixed-parameter LNS strategy lacks the dynamic response capability to distribution characteristics, weakening the algorithm's global exploration ability.

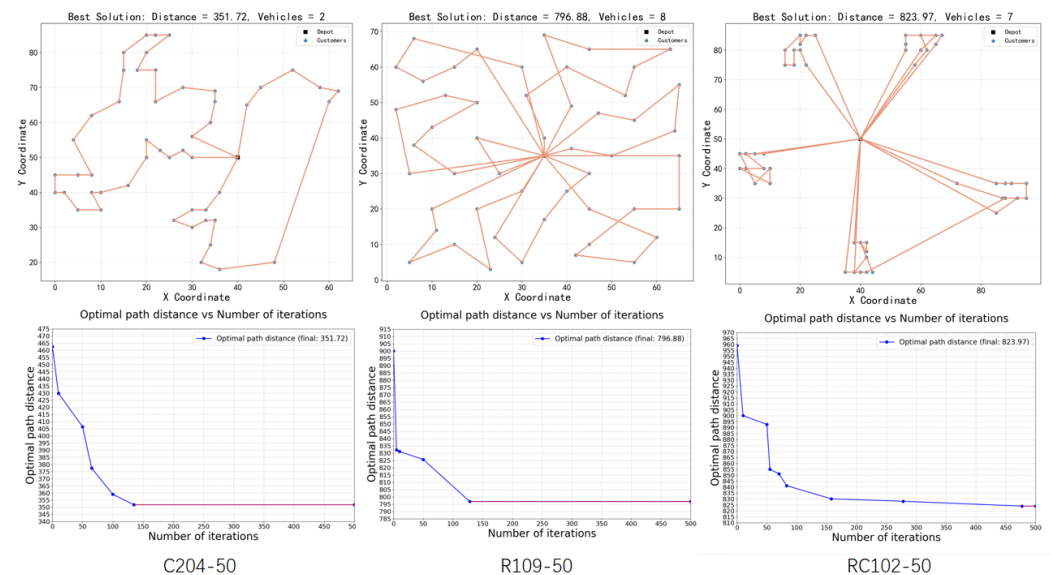


Figure 5. Partial Optimal Solution Route Map and Convergence Curve. The orange lines represent the vehicle's driving path, and the rose red lines represent its convergence situation.

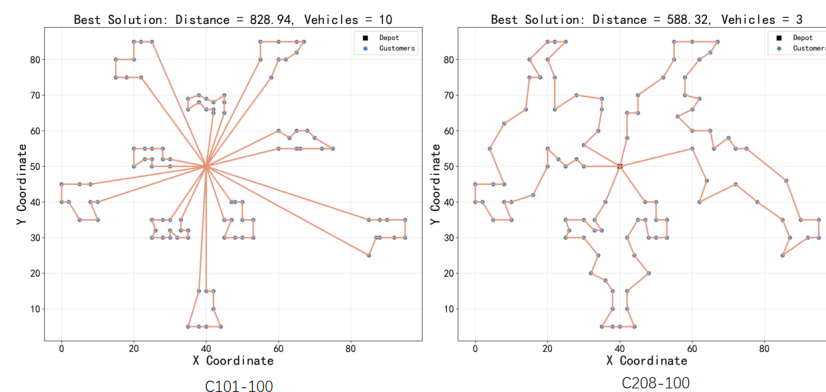


Figure 6. Route Map of Some Data Sets. The orange lines represent the vehicle's driving path.

6. Conclusions

This paper proposes a multi-objective collaborative framework based on the concept of symmetrical optimization, which combines the Grey Wolf Optimization algorithm, large neighborhood search, and the dual-thread ant colony algorithm to solve the vehicle routing problem with time windows (VRPTW). The framework achieves simultaneous optimization of multiple objectives, including the number of vehicles, travel costs, and time window violations, through a three-stage collaborative mechanism: initial solution construction, multi-objective balance optimization, and local fine search. In the initialization stage, an improved nearest neighbor heuristic algorithm is used to generate high-quality starting points. In the multi-objective optimization stage, the social hierarchy mechanism of the multi-objective grey wolf optimization is introduced to coordinate the exploration and

exploitation processes with symmetrical thinking. In the search stage, the dual-thread ant colony algorithm's collaborative mechanism is employed to simultaneously advance path optimization and vehicle number reduction as symmetrical objectives. In the dual-thread ant colony collaborative structure, the main thread ACS_Time focuses on optimizing path length and time window compliance, while the auxiliary thread ACS_vehicle aims to reduce the number of vehicles used. Both threads achieve symmetrical information exchange and objective collaboration through a shared pheromone matrix. By integrating an adaptive weight mechanism, elite strategy, and dynamic penalty function, the algorithm can effectively balance multiple objectives. Experimental results show that on the Solomon dataset, the proposed method can obtain a uniformly distributed Pareto solution set and significantly outperforms single-objective algorithms in terms of multi-objective comprehensive performance.

However, when the customer point scale reaches 100, the algorithm still has insufficient target balance ability in R and RC type instances. The future can build an adaptive optimization system from multiple dimensions: establish an initialization mechanism that perceives distribution characteristics, adopt a diversity enhancement strategy for R-type instances, and design a hybrid initialization method for RC-type instances; innovate a dynamic weight allocation scheme, strengthen the optimization weight of path cost in R-type instances, and balance the relationship among multiple objectives in RC-type instances; construct an intelligent parameter adjustment system, enabling the key parameters of LNS to adjust autonomously based on instance types and convergence states. Through the collaborative improvement of these three aspects, the algorithm's adaptability to complex distribution characteristics can be significantly enhanced, and its optimization stability and convergence efficiency in random and mixed scenarios can be improved.

Future research can be conducted in the aspect of application expansion. In terms of application, it is necessary to explore multi-modal transportation scenarios, extend the algorithm to complex scenarios such as drone and vehicle collaborative delivery and multi-vehicle type mixed scheduling, and reconstruct the solution and optimization objective expressions. As a core issue in logistics optimization, the research on the vehicle routing problem (VRPTW) needs to continuously deepen the alignment between algorithms and actual demands. Although the MOGWO-DTACO algorithm proposed in this study has made progress in multi-objective optimization, by integrating emerging technologies and enhancing dynamic adaptability, intelligent optimization algorithms will demonstrate greater value in intelligent logistics systems.

Author Contributions: Conceptualization, Y.Z., J.Y., P.C. and G.Z.; methodology, P.C.; software, Y.Z.; validation, Y.Z., J.Y., P.C. and G.Z.; formal analysis, P.C.; investigation, Y.Z.; resources, Y.Z.; data curation, J.Y. and G.Z.; writing—original draft preparation, Y.Z. and J.Y.; writing—review and editing, Y.Z. and P.C.; visualization, P.C.; supervision, Y.Z.; project administration, J.Y.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key Research and Development Program, grant number 2024YFC3809403.

Data Availability Statement: The data presented in this study are available in the OR-Library repository at <http://people.brunel.ac.uk/~mastijb/jeb/orlib/vrpinf.html> (accessed on 15 February 2025), reference number VRPTW-Solomon. These data were derived from the following resources available in the public domain: Solomon benchmark.

Conflicts of Interest: Yipeng Zhang and Junya Yang were employed by the CCCC Second Harbor Engineering Company Ltd. Pengyu Chen and Guoning Zhao were enrolled at the Wuhan University of Technology. The authors declare that this study received funding from National Key Research and Development Program, grant number 2024YFC3809403. The funder was not involved in the study

design, collection, analysis, interpretation of data, the writing of this article or the decision to submit it for publication.

References

1. Dantzig, G.B.; Ramser, J.H. The truck dispatching problem. *Manag. Sci.* **1959**, *6*, 80–91. [\[CrossRef\]](#)
2. Lenstra, J.K.; Kan, A.R. Complexity of vehicle routing and scheduling problems. *Networks* **1981**, *11*, 221–227. [\[CrossRef\]](#)
3. Si, J. Research on Mobile Robot Path Planning Based on Ant Colony Algorithm and Dynamic Window Approach. Master's Thesis, Shanghai Ocean University, Shanghai, China, 2024.
4. Fei, M.; Huang, D.; Lu, Y.; Qiao, J. Enhanced Gray Wolf Optimization Algorithm Integrating Multiple Improvement Methods. *J. Jilin Univ. (Sci. Ed.)* **2025**, *63*, 829–834. [\[CrossRef\]](#)
5. Menares, F.; Montero, E.; Paredes-Belmar, G.; Bronfman, A. A bi-objective time-dependent vehicle routing problem with delivery failure probabilities. *Comput. Ind. Eng.* **2023**, *185*, 109601. [\[CrossRef\]](#)
6. Li, X. Path planning of intelligent mobile robot based on Dijkstra algorithm. *J. Phys. Conf. Ser.* **2021**, *2083*, 042034. [\[CrossRef\]](#)
7. Li, C.; Huang, X.; Ding, J.; Song, K.; Lu, S. Global path planning based on a bidirectional alternating search A* algorithm for mobile robots. *Comput. Ind. Eng.* **2022**, *168*, 108123. [\[CrossRef\]](#)
8. LaValle, S.M.; Kuffner, J.J. Rapidly-exploring random trees: Progress and prospects. In *Algorithmic and Computational Robotics*; AK Peters/CRC Press: Natick, MA, USA, 2001; pp. 303–307.
9. Guan, L.; Bai, Y. Solving Traveling Salesman Problem Using Branch and Bound Algorithm. *J. North Univ. China (Nat. Sci. Ed.)* **2007**, *28*, 104–107.
10. Xu, Q.; Tang, G.; Gai, S.; Yang, X. Vectorized Path Planning Algorithm in Dynamic Environment. *Period. Ocean Univ. China* **2014**, *44*, 109–113. [\[CrossRef\]](#)
11. Shi, K.; Wu, Z.; Jiang, B.; Karimi, H.R. Dynamic path planning of mobile robot based on improved simulated annealing algorithm. *J. Frankl. Inst.* **2023**, *360*, 4378–4398. [\[CrossRef\]](#)
12. Zhang, T.W.; Xu, G.H.; Zhan, X.S.; Han, T. A new hybrid algorithm for path planning of mobile robot. *J. Supercomput.* **2022**, *78*, 4158–4181. [\[CrossRef\]](#)
13. Shi, Y.; Zhang, H.; Li, Z.; Hao, K.; Liu, Y.; Zhao, L. Path planning for mobile robots in complex environments based on improved ant colony algorithm. *Math. Biosci. Eng.* **2023**, *20*, 15568–15603. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Wang, Q. Collaborative Truck and UAV Delivery Route Optimization with Time Windows in Time-Dependent Networks. Master's Thesis, Dalian Maritime University, Dalian, China, 2024.
15. Pecin, D.; Pessoa, A.; Poggi, M.; Uchoa, E. Improved branch-cut-and-price for capacitated vehicle routing. *Math. Program. Comput.* **2017**, *9*, 61–100. [\[CrossRef\]](#)
16. Yu, Y.; Wang, S.; Wang, J.; Huang, M. A branch-and-price algorithm for the heterogeneous fleet green vehicle routing problem with time windows. *Transp. Res. Part B Methodol.* **2019**, *122*, 511–527. [\[CrossRef\]](#)
17. Xie, X.; Zhou, H.; Yang, Y. Application of Multi-Strategy Improved Particle Swarm Optimization Algorithm in VRPTW Problem. *Comput. Technol. Dev.* **2024**, *34*, 186–192. [\[CrossRef\]](#)
18. Li, H. Optimization of Vehicle Distribution Path with Time Windows Based on Improved Genetic Algorithm. *Bull. Sci. Technol.* **2025**, *41*, 36–41. [\[CrossRef\]](#)
19. Expósito, A.; Brito, J.; Moreno, J.A.; Expósito-Izquierdo, C. Quality of service objectives for vehicle routing problem with time windows. *Appl. Soft Comput.* **2019**, *84*, 105707. [\[CrossRef\]](#)
20. Guo, Q.; Dong, X.; Li, Q. Research on Solving VRPTW Using Genetic Algorithm Based on Adaptive Large Neighborhood Search. *J. Qingdao Univ. (Eng. Technol. Ed.)* **2023**, *38*, 1–9. [\[CrossRef\]](#)
21. Ghannadpour, S.F.; Zandiyeh, F. A new game-theoretical multi-objective evolutionary approach for cash-in-transit vehicle routing problem with time windows (A Real life Case). *Appl. Soft Comput.* **2020**, *93*, 106378. [\[CrossRef\]](#)
22. Huang, C.; Wu, Y. Research on Airport Terminal VRPTW Based on Multi-Objective Optimization Algorithm. *Sci. Technol. Ind.* **2025**, *25*, 58–63.
23. Sun, B.; Zhou, J.; Zhao, Y.; Zhang, Y.; Peng, H.; Zhao, W. Global Path Planning for Robot Based on Improved Gray Wolf Optimization Algorithm. *Sci. Technol. Eng.* **2024**, *24*, 14287–14297.
24. Chen, K.; Gong, Y. Hybrid Multi-Objective Gray Wolf Algorithm for Solving Multi-Objective VRPTW Problem. *Comput. Eng. Appl.* **2024**, *60*, 309–318.
25. Li, Y.; Huang, X.; Zhang, Z. Multi-Objective Path Planning Method Based on Fused Ant Colony and A* Algorithm. *Comput. Technol. Autom.* **2024**, *43*, 66–72. [\[CrossRef\]](#)
26. Wang, Y.; Wang, L.; Chen, G.; Cai, Z.; Zhou, Y.; Xing, L. An improved ant colony optimization algorithm to the periodic vehicle routing problem with time window and service choice. *Swarm Evol. Comput.* **2020**, *55*, 100675. [\[CrossRef\]](#)
27. Shen, Y.; Liu, M.; Yang, J.; Shi, Y.; Middendorf, M. A hybrid swarm intelligence algorithm for vehicle routing problem with time windows. *IEEE Access* **2020**, *8*, 93882–93893. [\[CrossRef\]](#)

28. Zhang, Y.; Li, J. A hybrid heuristic harmony search algorithm for the vehicle routing problem with time windows. *IEEE Access* **2024**, *12*, 42083–42095. [[CrossRef](#)]
29. Wang, Y.; Chen, X.; Shuang, Z.; Zhan, Y.; Chen, K.; Xu, C. Self-competition particle swarm optimization algorithm for the vehicle routing problem with time window. *IEEE Access* **2024**, *12*, 127470–127488. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.