

Article

An Adaptive Sliding-Window Ant Colony Optimization Approach for Multi-UAV Delivery Routing with Safety and Lateness Objectives

Xin-Yi Chen ¹, Lin Shi ² and Jian-Yu Li ^{3,*}¹ College of Software, Nankai University, Tianjin 300350, China; xinyichen@mail.nankai.edu.cn² School of Statistics and Data Science, Guangdong University of Finance & Economics, Guangzhou 510320, China; shilin@gdufe.edu.cn³ College of Artificial Intelligence, Nankai University, Tianjin 300350, China

* Correspondence: jianyuli@nankai.edu.cn

Abstract

Urban unmanned aerial vehicle (UAV) logistics systems require routing algorithms that can balance delivery efficiency and operational safety under complex physical and environmental constraints. However, existing multi-objective routing methods often show limited adaptability when the search landscape changes during optimization. In particular, conventional ant colony optimization (ACO) algorithms usually rely on fixed pheromone update rules, which may lead to premature convergence, uneven Pareto front distributions, and pheromone-induced search stagnation. To address these limitations, this paper proposes a Sliding-Window Adaptive Multi-Strategy Ant Colony Optimization (SW-MSACO) framework for complex urban UAV delivery routing. First, a Complex Multi-objective Urban Routing Problem (CMURP) is formulated by jointly considering cumulative delivery lateness and safety risk under payload and nonlinear battery energy constraints. Second, a multi-strategy pheromone evolution pool is designed to provide complementary search behaviors for convergence acceleration, safety-efficiency balancing, and Pareto front diversity preservation. Third, a Sliding-Window Adaptive Strategy Selection (SW-ASS) mechanism is introduced to adjust strategy selection probabilities according to recent search performance rather than long-term cumulative rewards. Comprehensive simulation experiments under different map scales and task-load conditions demonstrate that SW-MSACO achieves improved Pareto solution quality and search stability compared with existing heuristic optimization approaches, particularly under large-scale and high-load scenarios, confirming the effectiveness of the proposed framework for complex UAV logistics optimization.

Keywords: urban UAV logistics; multi-objective optimization; ant colony optimization; adaptive strategy selection; sliding-window mechanism; drone routing optimization; pareto optimization

Academic Editor: Sándor Szénási

Received: 28 May 2026

Revised: 2 July 2026

Accepted: 8 July 2026

Published: 11 July 2026

Copyright: © 2026 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](#).

1. Introduction

Urban drone logistics has emerged as a promising solution for next-generation last-mile delivery systems in smart cities [1]. Compared with traditional ground transportation, unmanned aerial vehicles (UAVs) offer significant advantages in delivery flexibility,

rapid response capability, and reduced dependence on road infrastructure [2–4]. However, despite these advantages, large-scale deployment of drone fleets in complex urban environments still faces substantial operational and computational challenges [5,6].

Unlike conventional vehicle routing problems, urban UAV routing must simultaneously consider multiple physical and environmental constraints. Limited battery endurance, payload capacity restrictions, delivery deadlines, and low-altitude safety risks collectively introduce strong coupling relationships among routing decisions. In particular, delivery efficiency and operational safety often conflict with each other. Consequently, urban drone routing inherently constitutes a complex multi-objective optimization problem involving both efficiency and safety trade-offs.

To address such challenges, numerous metaheuristic approaches have been introduced for UAV path planning and vehicle routing optimization, including Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) [7–10]. Among these approaches, ACO demonstrates strong advantages in distributed path construction and combinatorial optimization due to its positive feedback mechanism and probabilistic search behavior [11–13]. Nevertheless, conventional ACO frameworks typically rely on static pheromone update rules, which often suffer from premature convergence and limited adaptability under highly dynamic search landscapes. Furthermore, in multi-objective optimization scenarios, fixed update strategies may excessively emphasize either convergence speed or population diversity, leading to uneven Pareto front distributions or reduced exploration capability.

Recent studies have attempted to improve adaptive search behavior through Adaptive Operator Selection (AOS) mechanisms [14]. Although these approaches dynamically adjust strategy selection probabilities according to historical performance, they commonly depend on long-term cumulative rewards [15]. Such designs may introduce delayed credit assignment, causing the optimization process to become trapped within locally reinforced search patterns. In complex UAV routing environments characterized by nonlinear energy constraints and dynamically changing search demands, maintaining adaptability while avoiding search inertia remains a significant challenge.

To overcome these limitations, this paper proposes a Sliding-Window Adaptive Multi-Strategy Ant Colony Optimization (SW-MSACO) framework for the Complex Multi-objective Urban Routing Problem (CMURP). The proposed method establishes a multi-strategy pheromone evolution pool consisting of several complementary update operators with distinct search characteristics. Furthermore, a Sliding-Window Adaptive Strategy Selection (SW-ASS) mechanism is introduced to dynamically regulate strategy selection through a short-term memory buffer. By emphasizing recent search performance instead of long-term cumulative rewards, the proposed mechanism improves adaptability to dynamic optimization states while mitigating search stagnation. In addition, a simulation-based evaluation framework is integrated to explicitly model nonlinear UAV energy consumption and operational feasibility constraints.

The primary contributions of this study are summarized as follows:

First, in problem modeling, a Complex Multi-objective Urban Routing Problem (CMURP) is formulated for urban UAV logistics. The proposed model jointly considers delivery lateness and operational safety risk under payload and nonlinear battery energy constraints, enabling a more realistic representation of complex urban drone delivery environments.

Second, in pheromone evolution design, a multi-strategy pheromone evolution pool is proposed, where multiple complementary update operators are constructed to balance convergence acceleration and Pareto front diversity.

Third, in adaptive strategy management, a Sliding-Window Adaptive Strategy Selection (SW-ASS) mechanism is proposed, which utilizes a short-term memory buffer to

dynamically regulate the selection probabilities of different search strategies. By emphasizing recent search performance, the proposed mechanism mitigates pheromone-induced search stagnation and improves adaptability in complex routing environments.

By integrating the CMURP model, the multi-strategy pheromone evolution pool, and the SW-ASS mechanism, a novel Sliding-Window Adaptive Multi-Strategy Ant Colony Optimization algorithm (SW-MSACO) is developed for complex urban UAV routing optimization. The proposed framework provides an effective and adaptive solution for balancing efficiency and safety objectives under nonlinear operational constraints.

Comprehensive simulation experiments are conducted under various map scales and task-load conditions. The results demonstrate that the proposed SW-MSACO achieves competitive or superior performance compared with conventional fixed-strategy heuristics and hybrid approaches in terms of solution quality, robustness, and computational stability. These findings verify the effectiveness and scalability of the proposed framework in handling complex multi-objective UAV routing problems.

The remainder of this paper is organized as follows. Section 2 provides a brief overview of related work. Section 3 introduces the CMURP model. Section 4 details the proposed SW-MSACO framework, including the multi-strategy pheromone evolution pool and the SW-ASS mechanism. Section 5 presents the experimental setup and comparative performance analysis under different scenario configurations. Finally, Section 6 concludes the paper and discusses future research directions.

2. Background and Related Work

In recent years, unmanned aerial vehicle (UAV) logistics systems have received considerable attention for urban last-mile delivery. Early UAV routing studies mainly focused on single-objective optimization problems that aimed to minimize travel distance, delivery time, or operational cost [16]. With the growing complexity of urban logistics environments, research gradually shifted toward multi-objective routing frameworks incorporating delivery time windows, fleet coordination, and uncertain demand conditions [17–19].

Energy consumption modeling has become a critical component of UAV routing optimization because battery endurance directly determines route feasibility, fleet utilization, and the validity of routing schedules. Unlike conventional ground vehicles, UAV energy consumption depends not only on travel distance but also on payload, flight speed, mission profile, weather conditions, and operational maneuvers. Existing studies indicate that simplified distance-based or purely linear battery models may lead to inaccurate feasibility assessments, particularly under varying payloads or flight speeds. Thibbotuwawa et al. emphasized that factors such as payload, speed, weather, and flight status should be incorporated into UAV energy modeling [20] and further showed that routing configurations and operational parameters significantly influence energy consumption, highlighting the need to consider the interaction between route structure and UAV operating conditions [21]. In addition, experimental studies, such as that by Kierzkowski et al., demonstrate that non-cruise operations like holding maneuvers can also affect energy usage, with different maneuvering patterns and environmental conditions leading to varying power-consumption characteristics [22]. Overall, these findings suggest that practical UAV routing models should account for both nominal route energy consumption and additional energy costs arising from holding, waiting, wind disturbances, trajectory corrections, and airspace-management procedures.

Motivated by these findings, the present study incorporates a nonlinear payload- and speed-dependent energy-cost constraint into the CMURP model. Since the main focus of this work is the design and evaluation of the SW-MSACO optimization framework rather than high-fidelity aerodynamic system identification, the adopted energy model is formulated as a reduced-order simulation model. It is used to evaluate battery-feasibility

constraints consistently across all compared algorithms, while the limitations of this simplified energy representation are explicitly discussed in Section 5.5.

In parallel, safety-aware routing has become an important research direction. Existing studies incorporated environmental risk exposure, population density distribution, and urban no-fly zones into UAV routing optimization [23,24]. Such developments reflect growing concerns regarding low-altitude urban airspace safety and operational reliability. Consequently, modern UAV routing problems increasingly involve trade-offs between delivery efficiency and operational safety. However, many existing formulations treat time efficiency, energy consumption, and safety risk as separate or pairwise optimization concerns. Relatively limited attention has been devoted to jointly modeling the trade-off between delivery lateness and safety risk under both payload and nonlinear energy constraints.

To solve complex routing problems, metaheuristic optimization methods have been applied to UAV scheduling and path planning tasks. Evolutionary multi-objective algorithms such as NSGA-II have demonstrated effective optimization performance in drone routing and resource allocation problems [25]. In addition, swarm intelligence methods, particularly Ant Colony Optimization, have been widely adopted because of their distributed search behavior and natural suitability for combinatorial routing problems [11–13]. For ACO-based routing algorithms, pheromone update rules and heuristic information design play central roles in guiding solution construction and balancing exploration with exploitation. However, traditional ACO frameworks often use fixed update mechanisms, which may limit their adaptability when the search state changes during optimization [14].

To further enhance the adaptability of metaheuristics under dynamic search environments, Adaptive Operator Selection (AOS) and dynamic parameter control mechanisms have been widely investigated. Existing AOS frameworks typically employ multi-armed bandit (MAB) variants or reinforcement learning (RL) paradigms to adaptively allocate computational resources among competing search strategies based on their historical performance feedback [26]. Although these adaptive schemes can adjust selection probabilities over time, they commonly rely on long-term cumulative rewards or global statistical metrics to guide strategy evolution. Such long-term feedback may be less responsive to short-term changes in the search process, especially when the optimization landscape is highly constrained and dynamically reinforced by pheromone accumulation.

Despite these advances, several challenges remain. First, in dense urban environments, shortest delivery paths often involve higher safety risks, while overly conservative routing strategies may increase delivery delays. This trade-off has not been adequately addressed within existing multi-objective routing frameworks. Second, traditional ACO frameworks commonly employ fixed pheromone update mechanisms, which may gradually reinforce locally dominant search patterns and reduce exploration diversity under high-demand routing conditions. Finally, although adaptive metaheuristic mechanisms have been introduced to improve adaptability, many existing approaches rely heavily on long-term cumulative rewards and historical statistics when adjusting search strategies. Such designs may introduce delayed feedback and historical bias, causing premature convergence and reduced adaptability in dynamic optimization environments.

Motivated by these limitations, this study proposes a Sliding-Window Adaptive Multi-Strategy Ant Colony Optimization (SW-MSACO) framework for complex urban UAV routing optimization. The proposed method integrates a multi-strategy pheromone update mechanism with a sliding-window adaptive strategy selection framework to improve search adaptability while alleviating pheromone-induced search stagnation in nonlinear multi-objective routing environments.

3. CMURP Model

The Complex Multi-Objective Urban Routing Problem (CMURP) considered in this study is formulated as a multi-objective optimization model for urban UAV logistics systems. The proposed model simultaneously considers delivery efficiency and operational safety while maintaining routing feasibility in complex urban environments.

First, delivery efficiency is quantified by the cumulative delivery lateness across all served delivery nodes:

$$\min F_1(S) = \sum_{k=1}^K \sum_{i \in r_k} \max(0, t_i - d_i) \quad (1)$$

where K denotes the total number of drones, r_k represents the delivery route of drone k , t_i is the actual arrival time at node i , and d_i denotes the predefined delivery deadline.

Second, the total safety risk is defined by integrating the environmental risk encountered along each path segment:

$$\min F_2(S) = \sum_{k=1}^K \sum_{(i,j) \in r_k} R_{ij}^k \quad (2)$$

where R_{ij} represents the environmental risk associated with the path segment from node i to node j .

The two objectives exhibit an inherent trade-off relationship. Reducing delivery lateness generally encourages shorter routes, which may involve higher-risk regions in complex urban environments. In contrast, reducing safety risk may require detours around high-risk areas, potentially increasing delivery delays. Consequently, the CMURP requires an algorithm capable of identifying Pareto-optimal solutions [27] to support different operational preferences. Beyond the objective functions, two critical physical constraints must be satisfied to ensure mission feasibility:

First, the total weight of the cargo assigned to each drone must not exceed its maximum payload capacity C :

$$\sum_{i \in r_k} w_i \leq C, \quad \forall k \in \{1, \dots, K\} \quad (3)$$

Second, the energy consumption of each UAV route must remain within the maximum battery capacity E_{max} . To evaluate route-level battery feasibility, this study adopts a reduced-order nonlinear energy-cost model rather than a high-fidelity aerodynamic power model. For each route segment (i,j) , the energy cost is defined as:

$$E_{ij} = D_{ij} \left(\frac{C_0 + C_W W_{ij}}{V_{ij}} + C_V V_{ij} \right) \quad (4)$$

where D_{ij} denotes the travel distance between node i and node j , W_{ij} denotes the remaining payload carried by the UAV before traversing segment (i,j) , and V_{ij} denotes the cruising speed on this segment. The formulation allows V_{ij} to be specified for each route segment, so the proposed energy-cost model is not inherently restricted to fixed-speed operation. In the baseline simulation setting, a nominal cruising speed is adopted to maintain controlled comparisons among different routing algorithms. Unless otherwise specified, the nominal cruising speed is set to $V_0 = 5$.

The first term in the parentheses indicates that, under a given payload condition, lower speed may increase the time-related flight-maintenance cost per unit distance. The second term reflects the additional cost associated with higher-speed operation. Therefore, the model captures a nonlinear trade-off between payload-dependent endurance and speed-dependent energy expenditure.

The values of C_0 , C_W , C_V are selected as normalized simulation coefficients rather than experimentally calibrated aerodynamic constants while C and E_{max} are specified as normalized simulation parameters. Specifically, $C_0 = 100$ is used to represent the dominant baseline operating cost of the UAV, ensuring that even an unloaded flight segment

consumes a non-negligible amount of energy. $C_w = 1$ is used to introduce a payload-dependent increment, so that routes carrying heavier remaining payloads incur higher energy costs. $C_v = 0.1$ is used to preserve the speed-related penalty term in the reduced-order model. The maximum payload capacity C is set to be 20, and the initial battery capacity of each UAV is set to $E_{max} = 10,000$. The same energy-cost setting is applied to all compared algorithms to ensure a fair comparative evaluation.

For each sortie, the initial payload is calculated as the total weight of all orders assigned to that sortie. During simulation, the remaining payload is updated after each delivery, and the energy cost is accumulated along the sequence of visited nodes. If the accumulated energy cost exceeds the available battery capacity, the corresponding sortie is regarded as infeasible and is penalized during fitness evaluation.

It should be noted that this energy-cost model is designed for simulation-based route feasibility evaluation. It is not intended to reproduce the full aerodynamic power behavior of a specific UAV platform. High-fidelity aerodynamic effects, wind disturbance, battery aging, holding maneuvers, collision avoidance, and dynamic trajectory correction are not explicitly modeled in the current formulation. These factors may introduce additional energy consumption in real urban UAV operations and are therefore discussed as limitations in Section 5.5.

4. The Proposed SW-MSACO

To solve the complex bi-objective CMURP, this study proposes a Sliding-Window Adaptive Multi-Strategy Ant Colony Optimization (SW-MSACO) algorithm. Unlike traditional ACO algorithms that rely on static pheromone update rules, the proposed framework maintains a diverse pool of pheromone update strategies and dynamically adjusts their selection probabilities during the search process through a short-term memory feedback mechanism. The following subsections detail solution encoding, construction, the simulation-based evaluation, the multi-strategy pool, and the adaptive operator selection mechanism.

4.1. Solution Encoding

Given the complex assignment relationship among drones, orders, and sorties in the CMURP, we employ a hierarchical structure to represent a solution. A candidate solution S is defined as the aggregate set of flight paths for the entire drone fleet, denoted as $S = \{r_1, r_2, \dots, r_K\}$, where r_k represents the routing schedule for drone k . Due to strict constraints on battery capacity and payload, a single drone may be required to execute multiple sorties. Consequently, r_k is further decomposed into an ordered sequence of trips, $r_k = \{t_{k,1}, t_{k,2}, \dots, t_{k,d}\}$, where each trip consists of a continuous node sequence starting and ending at the depot.

4.2. Solution Construction

The Ant Colony System incrementally constructs routing solutions using both pheromone information and heuristic guidance. During each routing step, the algorithm probabilistically selects the next delivery node based on these two information sources. To ensure solution feasibility, the proposed SW-MSACO directly incorporates payload capacity constraints into the route construction process.

Unlike conventional implementations in which each ant constructs a single route, the proposed framework allows each ant to coordinate the routing schedules of the entire drone fleet. A round-robin scheduling mechanism is adopted to sequentially construct routes for different drones. Assuming a delivery scenario with N orders, the detailed construction procedure is summarized in Algorithm 1.

Algorithm 1: Solution Construction

Input: T —the pheromone matrix;
 H —the heuristic information matrix;
 K —drone fleet size;
 O —set of all unserved orders;
Output: $S = \{r_1, r_2, \dots, r_K\}$ —a complete delivery schedule;

- 1: **Begin**
- 2: Initialize an empty solution $S \leftarrow \emptyset$, the unserved order set $U \leftarrow O$, and designate the initial active drone $m \leftarrow 1$
- 3: **While** $U \neq \emptyset$ **Do**
- 4: Start a new trip $r_{m,new}$ at the depot and set current cargo weight $W_{current} \leftarrow 0$
- 5: **Repeat**
- 6: Construct the feasible neighborhood FN by filtering nodes in U that satisfy the capacity constraint
- 7: **If** $FN = \emptyset$ **Then**
- 8: Conclude the current trip and direct drone m back to the depot. Break the inner loop.
- 9: **Else**
- 10: Evaluate the transition likelihood for each $j \in FN$ based on T and H according to Equation (5)
- 11: Select the subsequent destination j^* via roulette wheel selection
- 12: Update state: $W_{current} \leftarrow W_{current} + w_{j^*}$, $r_{m,new} \leftarrow r_{m,new} \cup \{j^*\}$, and remove j^* from U
- 13: **End If**
- 14: **Until** the capacity or endurance limits are reached
- 15: Integrate the complete sortie $r_{m,new}$ into global schedule of drone m in S
- 16: Rotate the drone pointer: $m \leftarrow (m \bmod K) + 1$
- 17: **End While**
- 18: **End**

Step 1: Initialize the system. Initially, all UAVs are located at the central depot. Let $U \leftarrow O$ denote the set of initially unserved customer requests. Each ant maintains a location-tracking vector $L = \{l_1, l_2, \dots, l_K\}$, where l_k represents the current node of the k -th UAV.

Step 2: Build a trip for drone k . Each ant maintains the set of unassigned tasks U and uses an active pointer k to identify the UAV currently being routed. Route construction always starts from the depot. Unlike the classical traveling salesperson problem, the proposed method explicitly considers physical constraints during route construction. Suppose the k -th drone is currently at node i . It carries an accumulated payload of $W_{current}$. Before selecting the next destination, the algorithm filters the remaining nodes in U to generate a feasible neighborhood (FN). The FN contains only nodes that satisfy the payload constraint. Specifically, selecting any node from the FN does not cause the drone payload to exceed its maximum capacity.

If the FN contains at least one valid candidate, the ant selects a subsequent destination j . This decision process relies on a specific probabilistic transition mechanism. We denote the transition probability as P_{ij} . This metric represents the likelihood of moving from the current node i to the candidate node j . The exact mathematical formulation is defined as follows:

$$P_{ij}^k = \begin{cases} \frac{(\tau_{ij})^\alpha \cdot (\eta_{ij})^\beta}{\sum_{u \in FN_i^k} (\tau_{iu})^\alpha \cdot (\eta_{iu})^\beta}, & \text{if } j \in FN_i^k \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where τ_{ij} denotes the accumulated pheromone intensity along the edge (i, j) , with α determining its degree of influence. Correspondingly, η_{ij} represents the heuristic attractiveness of the segment, scaled by the importance factor β .

The heuristic component η_{ij} simultaneously considers Euclidean distance D_{ij} and environmental risk R_{ij} :

$$\eta_{ij} = \frac{1}{w_D \cdot D_{ij} + w_R \cdot R_{ij} + \epsilon} \quad (6)$$

In this expression, D_{ij} signifies the spatial distance separating the two nodes, and R_{ij} quantifies the operational risk assigned to the edge (i, j) . The parameters w_D and w_R act as weighting coefficients to balance travel efficiency against safety concerns. A small constant ϵ is added to prevent division by zero.

Based on the transition probabilities, roulette wheel selection is used to determine the next node. Upon confirming the target j , the ant loads the order onto the current drone, updates the drone's current payload $W_{current}$ and removes the served request from the global pending pool U . This stepwise extension of the route persists until no feasible candidate remains or the endurance limit is reached.

Step 3: Drone Switching. Eventually, the current sortie for drone k concludes. The drone returns to the depot when no additional feasible node can be inserted into the current trip. At this stage, the algorithm switches to the next UAV to improve workload balancing across the fleet. Specifically, the ant shifts its focus to the next drone in the sequence. Meanwhile, the ant resets its location back to the depot. A new trip construction process is then initiated. The algorithm iteratively builds trips and rotates the active drone until the set of unserved orders U becomes empty.

4.3. Solution Evaluation

In Section 4.2, each ant generates a constructed routing schedule for the entire UAV fleet. This schedule must then be translated into quantifiable operational metrics. To achieve this, a step-by-step simulation framework is introduced to emulate UAV flight behavior by updating all state variables according to the mathematical model defined in Section 3.

First, the generated solution is evaluated for feasibility. The payload constraint is primarily enforced during the route construction process in Section 4.2. However, battery consumption is nonlinear and therefore requires additional verification during post-construction simulation. For robustness, both payload and energy violations are still recorded during simulation. Any violation of these hard constraints renders the solution physically infeasible. Such an infeasible solution is denoted as S_{inf} . To discourage infeasible solutions during optimization, a penalty term is added to the fitness function as follows:

$$fit(S_{inf}) = \gamma \cdot (n_{cap} + n_{energy}) \quad (7)$$

where $fit(\cdot)$ denotes the fitness value of the solution, γ represents the prohibitive penalty coefficient (set to 10^9), and n_{cap} and n_{energy} denote the number of payload and energy constraint violations, respectively.

Subsequently, for feasible solutions, a different evaluation strategy is applied. The CMURP considers two objectives: cumulative delivery lateness and total safety risk. Since these objectives have different numerical scales, directly combining raw objective values may affect the balance of multi-objective evaluation. Therefore, a min-max normalization strategy is adopted:

$$fit(S) = w_1 \cdot \frac{F_1(S)}{lt_{max}} + w_2 \cdot \frac{F_2(S)}{rk_{max}} \quad (8)$$

where lt_{max} and rk_{max} denote the estimated upper bounds for lateness and risk, respectively. This normalization process ensures that delivery lateness and safety risk contribute on comparable numerical scales during fitness evaluation. In the current implementation, the weights are set to $w_1 = 1$ and $w_2 = 2$, where w_1 and w_2 correspond to delivery lateness and safety risk, respectively. This setting is adopted as a baseline safety-oriented preference configuration rather than a fixed operational policy. These weights can be adjusted to guide the search under different operational preferences. However, since SW-MSACO

returns a Pareto solution set rather than a single deterministic route, the weights should be interpreted as preference-guidance parameters for fitness evaluation and pheromone updating, rather than guarantees of strictly monotonic changes in lateness or risk.

4.4. The Multi-Strategy Pheromone Evolution Pool

To guide the search process toward high-quality regions of the decision space, pheromone trails are globally updated at the end of each iteration through evaporation and deposition:

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (9)$$

where ρ denotes the evaporation rate (set to 0.1), and $\Delta\tau_{ij}$ represents the deposited pheromone increment. The evaporation mechanism helps reduce excessive pheromone accumulation and maintain search diversity.

Given the complexity of multi-objective optimization, relying on a single pheromone update rule may not provide balanced convergence and diversity throughout different search stages. Therefore, a multi-strategy pheromone evolution pool consisting of three update operators is introduced.

4.4.1. Strategy I: Prioritizing Convergence (S_1)

Inspired by the MAX-MIN Ant System [28], this strategy utilizes only the best solution S_{best} , which has the minimum aggregated fitness value in the current iteration. During each iteration, pheromone deposition is applied exclusively to S_{best} :

$$\Delta\tau_{ij}(t) = \Delta\tau_{ij}^{best}, \quad \text{if } (i, j) \in S_{best} \quad (10)$$

By concentrating pheromone updates on the current best trajectory, S_1 improves convergence speed. However, the increased selection pressure may reduce population diversity. Therefore, this strategy is more suitable for scenarios requiring rapid convergence toward feasible high-quality solutions.

4.4.2. Strategy II: Balancing Safety and Efficiency (S_2)

In the proposed CMURP model, delivery efficiency and routing safety represent two competing objectives. Relying only on aggregated fitness values may bias the search toward one objective. To alleviate this issue, S_2 selects two elite solutions during each iteration: the fitness-optimal solution and the risk-optimal solution. The total pheromone increment $\Delta\tau_{ij}$ is then calculated as the combination of both contributions:

$$\Delta\tau_{ij}(t) = \Delta\tau_{ij}^{fit} + \Delta\tau_{ij}^{risk} \quad (11)$$

This dual-deposition mechanism preserves the attractiveness of safer path segments, even when such routes introduce additional travel distance. As a result, the search process maintains a balance between safety and delivery efficiency. Compared with single-objective pheromone reinforcement, this strategy helps prevent the search from being dominated only by the aggregated fitness value.

4.4.3. Strategy III: Enhancing Pareto Front Distribution (S_3)

A common challenge in multi-objective optimization is the uneven distribution of solutions along the Pareto front, where solutions may cluster in specific regions. To address this issue, the proposed method incorporates the Crowding Distance (CD) [29] metric from NSGA-II into the pheromone update process.

Specifically, fast non-dominated sorting is first performed on the current population to obtain the rank-1 non-dominated set PF . For each solution $k \in PF$, its crowding distance

CD_k is calculated according to the normalized distance between neighboring solutions in the objective space.

To support roulette-wheel-based probabilistic selection, a modified crowding distance assignment strategy is adopted. Since roulette wheel selection requires finite probability values, boundary solutions are assigned amplified but finite crowding distance values instead of infinite values. Specifically, each boundary solution is assigned twice the maximum finite crowding distance within the current non-dominated set. This mechanism increases the selection probability of boundary solutions while preserving opportunities for sparsely distributed internal solutions. The probability of selecting non-dominated solution k for pheromone deposition is defined as:

$$p_k = \frac{CD_k}{\sum_{m \in PF} CD_m} \quad (12)$$

4.5. Sliding-Window Adaptive Strategy Selection

Selecting an appropriate pheromone update strategy is a dynamic decision-making problem that depends on the current search state. During the optimization process, different strategies may exhibit different capabilities in balancing convergence and diversity preservation.

Adaptive Operator Selection (AOS) provides a natural framework for dynamically allocating search effort among competing strategies through performance-based probability adjustment. However, its integration into ACO faces two major challenges. First, strategy contributions may exhibit delayed feedback, where the influence of a strategy on global solution quality becomes observable only after several iterations. Second, pheromone-induced search inertia may occur when early reinforcement of specific search patterns reduces subsequent exploration diversity.

To address these issues, a Sliding-Window Adaptive Strategy Selection (SW-ASS) mechanism is proposed. The proposed mechanism introduces a short-term memory buffer to reduce the influence of long-term historical bias on current strategy selection. The execution procedure of SW-ASS is summarized in Algorithm 2. The mechanism consists of three main components: strategy performance evaluation, sliding-window memory management, and probabilistic strategy selection.

Algorithm 2: SW-ASS

Input: W —the sliding-window buffer;
 S —the strategy pool;
 P_t —current probability distribution of strategies in W ;
 S_{prev} —the strategy selected in the previous iteration;

Output: S_m —the strategy selected for the current iteration;
 P_{t+1} —updated probability for strategies in W

- 1: **Begin**
- 2: Select the current strategy S_m via roulette wheel selection based on P_t
- 3: Evaluate S_{prev} using the solution quality observed in the current iteration:
- 4: **If** S_{prev} identifies a new global optimal fitness in the next iteration $t + 1$ **Then**
- 5: $R_{con} \leftarrow 1.0$
- 6: **End If**
- 7: **If** S_{prev} contributes more than two new solutions to Pareto archive **Then**
- 8: $R_{div} \leftarrow 0.5$
- 9: **End If**
- 10: Calculate composite reward: $R(t) \leftarrow R_{con} + R_{div}$
- 11: Push the performance tuple $(S_{prev}, R(t))$ into the rear of window W
- 12: **If** count(W) > 10 **Then**

```

13:     Pop the oldest entry from the front of  $W$ 
14: End If
15: For every strategy  $S_j \in S$  Do
16:     Refresh  $P_{t+1}(S_j)$  by normalizing rewards according to Equation (14)
17: End For
18: Return updated probability vector  $P_{t+1}$ 
19: End

```

4.5.1. Strategy Performance Evaluation

To quantitatively evaluate the effectiveness of different pheromone update strategies, let S_m denote the strategy selected at iteration $t-1$. The performance of S_m is evaluated according to the quality of the solutions generated in the subsequent iteration t . This delayed-feedback design allows the algorithm to assess the actual search effect of the previously selected strategy before updating the strategy probabilities.

Subsequently, a composite reward function $R(S_m, t)$ is introduced to measure the contribution of each strategy to two objectives of multi-objective optimization: convergence improvement and Pareto front diversity.

Specifically, a convergence reward $R_{con} = 1.0$ is assigned if the population at iteration t identifies a new global best fitness value, indicating that the pheromone update associated with S_m in the previous step effectively guided the ants toward superior regions of the decision space. Simultaneously, a credit of 0.5 is allocated to R_{div} provided that the strategy results in more than two new non-dominated solutions being added into the global Pareto archive.

Consequently, the total reward for the current iteration is defined as:

$$R(S_m, t) = R_{con}(S_m, t) + R_{div}(S_m, t) \quad (13)$$

which is subsequently archived in the memory buffer to drive the dynamic adjustment of strategy selection probabilities, as detailed in the following sections.

The reward function is designed as an event-based feedback signal rather than a continuous quality indicator. Continuous indicators such as HV improvement, IGD, spacing, or the epsilon indicator can provide more detailed performance information, but they also require additional reference points, normalization procedures, or representative Pareto reference sets. In particular, HV improvement is sensitive to the reference-point setting, while IGD requires a reference Pareto set that is generally unavailable during online optimization. Therefore, the current implementation adopts two directly observable search events, namely global-best improvement and Pareto-archive expansion, to provide a lightweight and stable feedback signal for online adaptive strategy selection.

4.5.2. Sliding-Window Memory Buffer

To facilitate adaptive strategy selection, the proposed algorithm maintains a short-term memory structure referred to as the sliding-window memory buffer W . Unlike traditional adaptive schemes that accumulate performance information throughout the entire optimization process, this mechanism adopts a first-in-first-out (FIFO) queue with a fixed capacity of 10 iterations.

At the conclusion of each iteration t , the reward value $R(S_m, t)$ obtained in Section 4.5.1 is pushed into the buffer, while the chronologically oldest entry is systematically removed to maintain the fixed window length.

The primary function of this buffer is to introduce recency-sensitive adaptation into the strategy selection process. By restricting the evaluation to recent interactions between the strategy pool and the search process, the algorithm reduces excessive dependence on long-term historical performance. This mechanism prevents strategies that perform well

during early search stages from continuously dominating later search stages, thereby maintaining adaptation capability under dynamic optimization conditions.

4.5.3. Probabilistic Strategy Selection

Based on the accumulated rewards stored in the memory buffer, the selection probability $P(S_m, t + 1)$ for each strategy is updated at the end of iteration t as follows:

$$P(S_m, t + 1) = \frac{\sum_{i \in W'} R(S_m, i) + \delta}{\sum_{j \in S} \left(\sum_{i \in W'} R(S_j, i) + \delta \right)} \quad (14)$$

where $i \in W$ denotes the set of iterations currently contained in the sliding window, and S represents the available strategy pool. A smoothing constant $\delta = 0.1$ is incorporated as a lower-bound preservation term [30]. This constant helps avoid strategy starvation, where the selection probability of a strategy becomes excessively small because of temporary underperformance. Consequently, each strategy retains the opportunity to participate in future search stages.

In addition, the proposed probabilistic framework helps alleviate search stagnation caused by local optima. When the algorithm fails to identify improved solutions over consecutive iterations, the accumulated reward terms $\sum R_{i,m}$ gradually decrease within the sliding window. Under such conditions, the smoothing constant δ ensures that the selection probabilities converge to a uniform distribution

$$P(S_m) \rightarrow \frac{1}{|S|} \quad (15)$$

As a result, the algorithm increases exploration across different search strategies, improving the possibility of escaping local optima.

Finally, the active strategy S_m is selected through roulette wheel sampling according to the updated probabilities

From a theoretical perspective, the sliding-window buffer acts as a finite-memory reward estimator. Unlike cumulative-reward adaptive selection mechanisms, which aggregate strategy performance over the entire optimization process, the proposed SW-ASS evaluates each strategy according to its recent contribution within the current window. As the window moves forward, outdated reward information is gradually discarded. Therefore, strategies that were effective only in early iterations cannot permanently dominate the selection probability if their recent performance becomes weak.

Meanwhile, the use of a finite memory window does not mean that all long-term search information is discarded. In the proposed ACO framework, accumulated search experience is still preserved by the pheromone matrix through evaporation and deposition. In addition, the smoothing constant (δ) in the probability update rule guarantees that each strategy retains a non-zero probability of being selected, preventing strategy starvation. Therefore, the sliding-window mechanism balances short-term adaptability and long-term exploration capability. It should be interpreted as a finite-memory adaptive control mechanism that reduces the excessive influence of outdated rewards without completely sacrificing the long-term search behavior of the underlying ACO process.

4.6. The Complete SW-MSACO

This section presents the complete SW-MSACO framework. Figure 1 illustrates the overall workflow of the proposed algorithm, while Algorithm 3 provides its detailed pseudocode implementation. The SW-MSACO is designed as a closed-loop evolutionary system in which search behavior and strategy adaptation are tightly coupled.

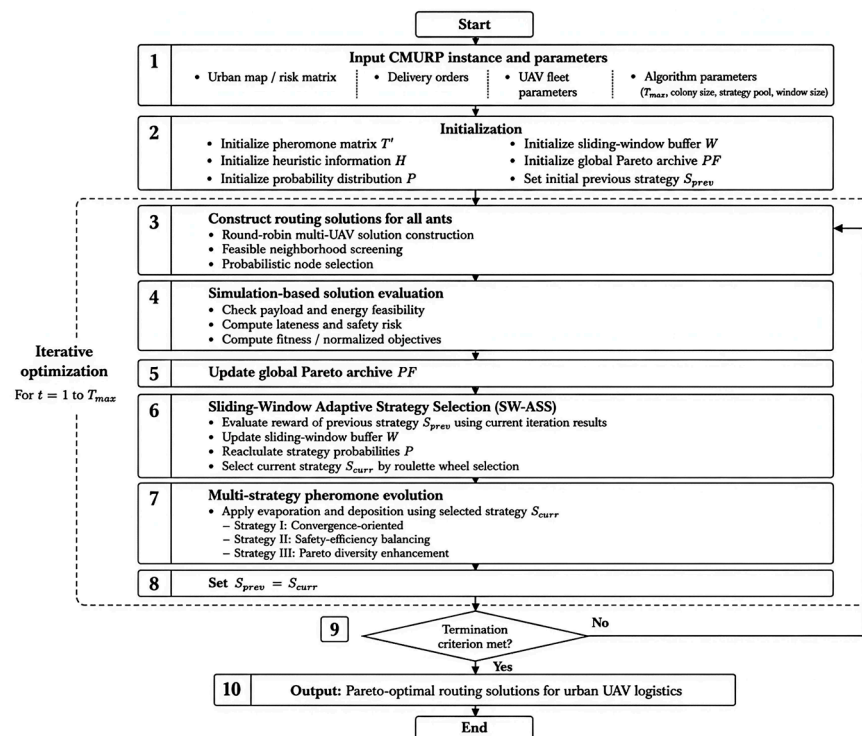


Figure 1. Overall workflow of the proposed SW-MSACO algorithm. The numbered arrows indicate the main execution sequence, while the feedback arrows represent the adaptive strategy-selection loop.

At the initiation of the optimization process, pheromone trails and heuristic information are initialized to provide baseline guidance for the UAV fleet. Simultaneously, the memory buffer W is reset, and the strategy selection probabilities p are assigned a uniform distribution to ensure an unbiased start for the strategy pool. This initialization establishes the necessary computational environment for the subsequent iterative loop, as shown in the upper part of Figure 1.

The core of each generation begins with the solution construction phase. Guided by the pheromone inherited from the preceding generation, a population of ants coordinates the UAV fleet to incrementally construct routing schedules S through the round-robin process formalized in Algorithm 1. These candidate solutions are then processed by a simulation-based evaluation module, which verifies physical feasibility and quantifies the performance regarding lateness and risk. Consequently, the global Pareto archive is updated with the newly identified non-dominated solutions. These steps correspond to the main iterative optimization loop in Figure 1.

Following evaluation, the algorithm executes the adaptive strategy selection (SW-ASS) phase. The quality of the solutions constructed in the current iteration is utilized to reward the pheromone update strategy $S_m(t - 1)$ selected in the preceding iteration. In Algorithm 3, S_{prev} records the strategy applied in the previous iteration and is used for delayed reward assignment in the next iteration. This performance feedback is registered in the sliding window W to drive the recalibration of the probability distribution p . Based on these updated probabilities, a roulette wheel selection is performed to designate the active strategy $S_m(t)$ for the current iteration.

In the final stage of each iteration, the pheromone matrix is updated according to the selected strategy $S_m(t)$. Specifically, one of the three pheromone evolution strategies is applied to guide subsequent solution construction: convergence-oriented reinforcement, safety-efficiency balancing, or Pareto diversity enhancement. This interaction between SW-ASS and multi-strategy pheromone evolution forms the adaptive closed-loop search

mechanism of SW-MSACO. The iterative process repeats until the termination criterion T is satisfied, after which the final Pareto set is returned.

Algorithm 3: The Complete SW-MSACO

Input: T_{max} —max iterations;
 N —the colony size;
 S —the strategy pool
 W —the window capacity

Output: PF —the global Pareto archive

- 1: **Begin**
- 2: Initialize pheromone matrix T , heuristic information H , memory buffer $W \leftarrow \emptyset$, and probabilities $P \leftarrow \frac{1}{|S|}$
- 3: Designate an initial strategy S_{prev} for the first iteration's setup.
- 4: **For** iteration $t = 1$ to T_{max} **Do**
- 5: **For** every ant $n = 1$ to N **Do**
- 6: Construct a fleet-wide routing solution S_n via **Algorithm 1**.
- 7: **End For**
- 8: Execute step-by-step simulation to verify constraints and quantify objectives.
- 9: Update the global Pareto archive PF with new non-dominated solutions.
- 10: Select pheromone update strategy S_{curr} adaptively based on $\{S_n\}$, W , P and S_{prev} according to **Algorithm 2**.
- 11: Apply pheromone evaporation and deposition using the rules of S_{curr} .
- 12: $S_{prev} \leftarrow S_{curr}$.
- 13: **End For**
- 14: **End**

4.7. Computational Complexity Analysis

To further clarify the scalability of the proposed SW-MSACO, this subsection analyzes its computational complexity. Let T_{max} denote the maximum number of iterations, N the number of ants, M the number of delivery orders, K the number of UAVs, A the size of the global Pareto archive, P the size of the current non-dominated subset, $|S|$ denote the number of pheromone update strategies, and W denote the sliding-window capacity.

In each iteration, solution construction requires $O(NM^2)$ since each ant repeatedly scans the remaining feasible orders during route construction. The simulation-based evaluation requires $O(NM)$, because each constructed solution visits each order once. Updating the global Pareto archive requires $O((A + N)^2)$ in the worst case when pairwise dominance comparison is used. When the diversity-oriented strategy is selected, crowding-distance calculation introduces an additional $O(P \log P)$ cost. The pheromone update requires $O(M^2)$ if the pheromone matrix is fully evaporated, since all pairwise pheromone entries between orders need to be updated. The sliding-window probability update in SW-ASS requires $O(|S|W)$, because it scans the recent W records for each of the $|S|$ strategies to update their selection probabilities.

Therefore, the overall worst-case time complexity of SW-MSACO is $O(NM^2 + NM + (A + N)^2 + P \log P + M^2 + |S|W)$.

Compared with fixed-strategy ACO, SW-MSACO preserves the same dominant complexity terms for solution construction, evaluation, Pareto archive maintenance, and pheromone updating. The only additional cost is the sliding-window probability update $O(|S|W)$. In this study, $|S| = 3$ and $W = 10$ so this overhead is effectively constant. Compared with S-Hybrid, SW-MSACO also has the same asymptotic order, except that S-Hybrid uses a static strategy combination and does not perform sliding-window probability recalculation. Thus, the adaptive mechanism of SW-MSACO does not change the dominant computational complexity; the main computational burden still comes from solution construction and Pareto archive maintenance.

5. Experimental Studies

5.1. Experimental Setup

All optimization and evaluation procedures are implemented within a unified simulation framework. The modeling, simulation, and algorithm components are fully developed and evaluated in MATLAB R2024b.

To evaluate the robustness, computational efficiency, and scalability of the proposed SW-MSACO algorithm under complex constraints, a series of simulation experiments was conducted.

First, to systematically investigate algorithmic performance under different geographical layouts and task-load conditions, a two-phase test suite comprising 11 distinct scenarios is constructed. These scenarios are organized along two dimensions: map scale and task load.

Next, to quantitatively assess the quality of the obtained Pareto solution sets, the hypervolume (HV) [31,32] metric is adopted. HV is widely used in multi-objective optimization because it simultaneously reflects the convergence and diversity of the obtained Pareto solutions with respect to a reference point.

However, since the risk index and lateness duration vary significantly in magnitude across different scenario configurations, the use of a globally fixed reference point may obscure local convergence characteristics and potentially lead to invalid HV evaluations. Therefore, a scenario-specific dynamic reference point is designed for each instance [33,34]. To avoid introducing algorithm-specific bias, the reference point is not determined separately for each algorithm. Instead, for each scenario, the final Pareto solutions generated by all compared algorithms over all independent runs are first collected, and a common reference point is then calculated from the worst objective bounds of this pooled solution set. The same reference point is then applied to all algorithms within the same scenario when calculating HV values.

To mitigate the impact of the stochastic uncertainty inherent in metaheuristic algorithms, each strategy is independently executed twenty times [35] for each scenario. The mean HV, standard deviation (Std. Dev.) and average runtime (Time) are recorded as the final evaluation metrics. To further evaluate whether the observed performance differences are statistically significant, a paired Wilcoxon signed-rank test is conducted between SW-MSACO and each competing baseline based on the HV values obtained from the 20 independent runs. The significance level is set to ($\alpha=0.05$). Since stochastic metaheuristic algorithms do not necessarily satisfy the normality assumption, this non-parametric test provides a more robust statistical comparison of the paired performance distributions.

5.2. Robustness and Efficiency Analysis

As described in Section 5.1, to rigorously evaluate the optimization robustness and computational efficiency of the proposed SW-MSACO, comprehensive comparative experiments are conducted against three fixed single heuristics (S_1 , S_2 and S_3) and a static hybrid baseline (S-Hybrid) [36]. The experimental analysis is performed from the perspectives of spatial scalability and robustness under increasing combinatorial density. Here, combinatorial density resilience refers to the algorithm's capability to maintain operational stability when handling highly overlapping delivery requests within constrained network layouts.

5.2.1. Robustness and Efficiency Under Map Scale Expansion

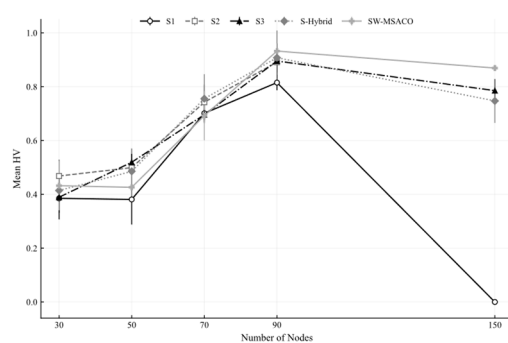
This experimental track evaluates the algorithm's capability to maintain search efficiency and solution quality as the physical decision space expands from a 30-node scenario to a large-scale 150-node scenario while maintaining a constant workload of 50

delivery orders. The corresponding experimental results are presented in Table 1 and Figure 2.

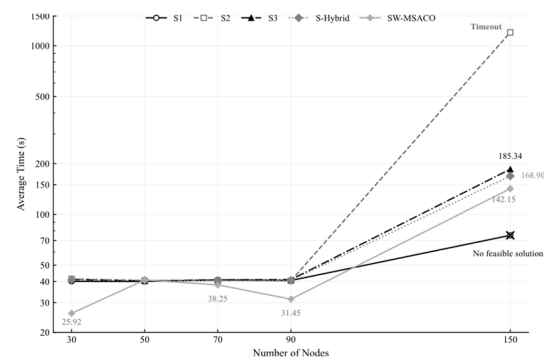
Table 1. Hypervolume Performance and Wilcoxon Signed-Rank Test Results under Map-Scale Expansion.

Case	Nodes	S ₁	S ₂	S ₃	S-Hybrid	SW-MSACO
Case 1	30	0.3852 ± 0.0534	0.4681 ± 0.0612	0.3891 ± 0.0821	0.4143 ± 0.0412	0.4321 ± 0.0914
Wilcoxon <i>p</i> -value		0.2188 (=)	0.0313 (−)	0.3438 (=)	0.4063 (=)	--
Case 2	50	0.3811 ± 0.0934	0.4992 ± 0.0712	0.5193 ± 0.0311	0.4854 ± 0.0211	0.4261 ± 0.0461
Wilcoxon <i>p</i> -value		0.1563 (=)	0.0274 (−)	0.0313 (−)	0.0431 (−)	--
Case 3	70	0.7012 ± 0.0341	0.7423 ± 0.0652	0.6961 ± 0.0212	0.7554 ± 0.0911	0.6921 ± 0.0912
Wilcoxon <i>p</i> -value		0.4375 (=)	0.1875 (=)	0.5123 (=)	0.0938 (=)	--
Case 4	90	0.8152 ± 0.0285	0.8911 ± 0.0382	0.8954 ± 0.0653	0.9077 ± 0.1012	0.9324 ± 0.0287
Wilcoxon <i>p</i> -value		0.0054 (+)	0.0241 (+)	0.0313 (+)	0.0185 (+)	--
Case 5	150	0.0000 ^a ± 0.0000	N/A ^b	0.7854 ± 0.0424	0.7471 ± 0.0817	0.8692 ± 0.0121
Wilcoxon <i>p</i> -value		0.0012 (+)	0.0051 (+)	0.0210 (+)	0.0143 (+)	--

Bold values indicate the best valid result for each metric within each case. The first number is Mean HV, where larger values are better, and the second number is Std. Dev., where smaller values are better. The Wilcoxon rows report the paired Wilcoxon signed-rank test results between SW-MSACO and each corresponding baseline based on 20 independent runs. The symbols +, −, and = indicate that SW-MSACO is significantly better than, significantly worse than, or not significantly different from the corresponding baseline at the significance level of ($\alpha = 0.05$), respectively. The dash “--” denotes that the comparison with SW-MSACO itself is not applicable. ^a No feasible solution was generated under the corresponding scenario constraints. ^b The algorithm exceeded the maximum runtime threshold without producing valid evaluation results.



(a) Mean HV with Std. Dev.



(b) Average runtime. Dashed annotations indicate timeout or infeasible-solution cases.

Figure 2. Performance comparison under map scale expansion.

According to Table 1 and Figure 2a, SW-MSACO does not universally dominate the fixed strategies in small-scale environments. In Cases 1–3, several fixed or static hybrid strategies achieve comparable or higher mean HV values. The Wilcoxon signed-rank test further confirms that most differences in these small-scale cases are not statistically significant, while SW-MSACO is significantly inferior to S₂ in Case 1 and to S₂, S₃, and S-Hybrid in Case 2. This observation is reasonable because the search space is relatively limited, and the adaptive strategy selection mechanism may not yet provide a clear advantage over simpler fixed update rules. Nevertheless, SW-MSACO still demonstrates competitive performance in terms of solution quality and an efficiency advantage in specific cases. As shown in Figure 2b, in Case 1, SW-MSACO reduces the execution time to 25.92 s, representing a 37.4% reduction compared with the 41.45 s required by S₂.

As the map scale expands to 90 and 150 nodes, the advantages of SW-MSACO become increasingly evident. In Case 4 (90 nodes), SW-MSACO achieves the highest mean HV of 0.9324 with a low standard deviation of 0.0287. More importantly, the Wilcoxon results indicate that SW-MSACO significantly outperforms all four baselines in this scenario, with $p = 0.0054$, 0.0241, 0.0313, and 0.0185 against S1, S2, S3, and S-Hybrid, respectively. Meanwhile, Figure 2b shows that its execution time is reduced to 31.45 s, which is lower than that of all competing strategies.

When the environment reaches the spatial scale of 150 nodes, conventional strategies begin to exhibit more substantial performance degradation. Specifically, S1 fails to generate feasible solutions under the corresponding scenario constraints, while S2 exceeds the maximum runtime threshold without producing valid evaluation results. This infeasibility and timeout behavior are also marked in Figure 2b.

In contrast, SW-MSACO maintains relatively high computational efficiency, requiring only 142.15 s. Moreover, it achieves the highest solution quality among all valid compared methods, with a mean HV of 0.8692 and a low fluctuation level (Std. Dev. = 0.0121). The Wilcoxon results also show statistically significant advantages over all comparable baselines in Case 5. These results indicate that the proposed sliding-window mechanism can effectively reduce the influence of misleading historical search bias. Furthermore, it enables the drone swarm to alleviate large-scale local stagnation under extreme spatial constraints.

5.2.2. Robustness and Efficiency Under Task Load Stress

This experimental track fixes the spatial network scale at 80 nodes while progressively increasing the delivery workload from 40 to 200 orders. Under this constrained network layout, the increasing concentration of delivery requests intensifies time-window overlaps and path-coupling conflicts, thereby imposing substantial combinatorial pressure on the routing system. The corresponding experimental results are presented in Table 2 and Figure 3.

Table 2. Hypervolume Performance and Wilcoxon Signed-Rank Test Results under Task Load Stress.

Case	Orders	S ₁	S ₂	S ₃	S-Hybrid	SW-MSACO
Case 6	40	0.7534 ± 0.3495	0.8172 ± 0.1332	0.8950 ± 0.0765	0.7816 ± 0.1780	0.7713 ± 0.1239
Wilcoxon p -value		0.5938 (=)	0.2813 (=)	0.0313 (−)	0.3438 (=)	--
Case 7	70	0.5654 ± 0.0874	0.6199 ± 0.0920	0.6247 ± 0.0510	0.6038 ± 0.0541	0.6175 ± 0.0715
Wilcoxon p -value		0.2188 (=)	0.1563 (=)	0.0442 (−)	0.2500 (=)	--
Case 8	100	0.3137 ± 0.0138	0.3180 ± 0.0211	0.2903 ± 0.0298	0.3109 ± 0.0465	0.3058 ± 0.0174
Wilcoxon p -value		0.0938 (=)	0.0625 (=)	0.6875 (=)	0.1250 (=)	--
Case 9	130	0.2395 ± 0.0138	0.2231 ± 0.0069	0.2507 ± 0.0117	0.2379 ± 0.0029	0.2603 ± 0.0086
Wilcoxon p -value		0.0253 (+)	0.0142 (+)	0.1013 (=)	0.0177 (+)	--
Case 10	160	0.1932 ± 0.0322	0.1683 ± 0.0069	0.1917 ± 0.0214	0.1852 ± 0.0091	0.2171 ± 0.0039
Wilcoxon p -value		0.0122 (+)	0.0085 (+)	0.0274 (+)	0.0091 (+)	--
Case 11	200	0.0000 ^a ± 0.0000	N/A ^b	0.1263 ± 0.0610	0.1395 ± 0.0136	0.1489 ± 0.0013
Wilcoxon p -value		0.0039 (+)	0.0042 (+)	0.0154 (+)	0.0108 (+)	--

Bold values indicate the best valid result for each metric within each case. The first number is Mean HV, where larger values are better, and the second number is Std. Dev., where smaller values are better. The Wilcoxon rows report the paired Wilcoxon signed-rank test results between SW-MSACO and each corresponding baseline based on 20 independent runs. The symbols +, −, and = indicate that SW-MSACO is significantly better than, significantly worse than, or not significantly different from the corresponding baseline at the significance level of ($\alpha = 0.05$), respectively. The dash “--”

denotes that the comparison with SW-MSACO itself is not applicable. ^a No feasible solution was generated under the corresponding scenario constraints. ^b The algorithm exceeded the maximum runtime threshold without producing valid evaluation results.

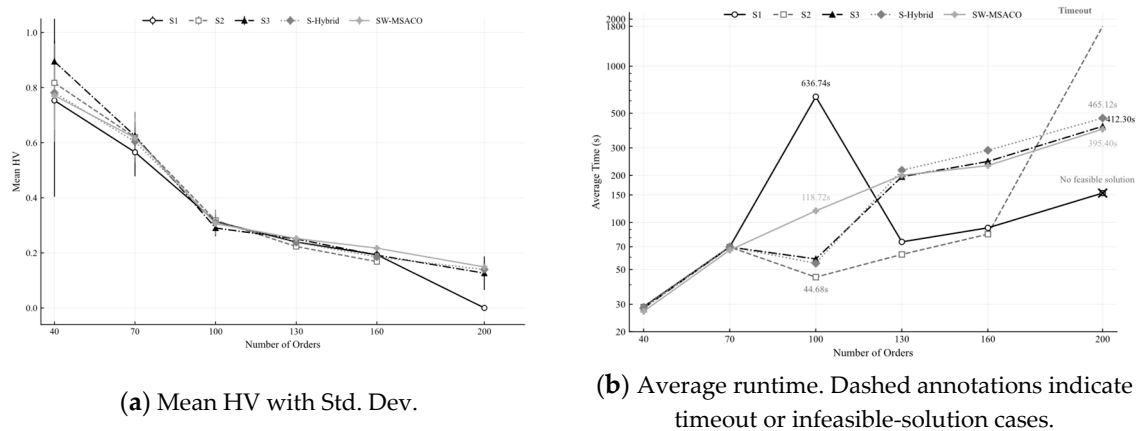


Figure 3. Performance comparison under task load stress.

The results in Table 2 and Figure 3a indicate that increasing task loads lead to substantial combinatorial complexity for all compared strategies. In low- and medium-density scenarios (Cases 6–8), SW-MSACO does not always achieve the highest HV value. Wilcoxon results show that most differences are not statistically significant. These observations suggest that when the task load is moderate, fixed update strategies may still be effective because the search landscape is not sufficiently complex to fully benefit from adaptive strategy selection.

A notable transition is observed in Case 8 (100 orders). Although the HV differences among competing strategies remain relatively small, Figure 3b shows that the execution time of S1 increases to 636.74 s, indicating that fixed convergence-oriented heuristics begin to exhibit computational instability under increasing task-coupling pressure.

As the workload further increases toward high-density conditions (Cases 9–11), the advantages of SW-MSACO become more evident.

Specifically, in the 130-order scenario (Case 9), SW-MSACO achieves the highest-quality Pareto frontier with a mean HV of 0.2603 while maintaining a low variance (Std. Dev. = 0.0086). In the 160-order scenario (Case 10), SW-MSACO further achieves the highest mean HV of 0.2171, outperforming all fixed and hybrid baselines. Under the maximum workload condition of 200 orders (Case 11), S-Hybrid degrades to an HV value of 0.1395 while SW-MSACO achieves the best mean HV of 0.1489 together with a highly stable standard deviation (Std. Dev. = 0.0013). The corresponding Wilcoxon results again confirm advantages over all comparable baselines. Meanwhile, in terms of computational performance, as shown in Figure 3b, S1 fails to generate a feasible solution, whereas S2 reaches the timeout condition. Compared with S3, SW-MSACO improves solution quality by 17.9% while maintaining acceptable computational efficiency (395.40 s).

Overall, the task-load stress experiment demonstrates that the proposed SW-MSACO is not designed to simply dominate fixed strategies in all low-complexity cases. Instead, its main advantage emerges under high task-load conditions, where delivery requests become highly coupled and the search landscape changes more dynamically. The statistically significant improvements in Cases 10–11, together with the partial significant advantages observed in Case 9, support the effectiveness of the sliding-window adaptive feedback mechanism in mitigating stagnation and maintaining stable Pareto-front quality under dense urban UAV delivery workloads.

5.3. Ablation Analysis of SW-ASS

To further examine the contribution of the proposed sliding-window adaptive strategy selection mechanism, an ablation experiment was conducted on four representative complex scenarios. The pheromone strategy pool was kept unchanged, while the strategy-selection mechanism was replaced by three alternatives: static hybrid selection, uniform random selection, and cumulative reward selection. The comparison results are reported in Table 3.

Table 3. Ablation results of different strategy-selection mechanisms.

Method	Mean HV $\pm$ Std. Dev. (Wilcoxon signed-rank)			
	Case 5	Case 9	Case 10	Case 11
SW-ASS	0.8231 $\pm$ 0.046	0.2847 $\pm$ 0.0136	0.2182 $\pm$ 0.0114	0.1679 $\pm$ 0.0013
Cumulative reward	0.8067 $\pm$ 0.0238 (+)	0.2973 $\pm$ 0.0292 (=)	0.1937 $\pm$ 0.0915 (+)	0.1435 $\pm$ 0.0184 (+)
Static hybrid	0.7945 $\pm$ 0.0392 (+)	0.2812 $\pm$ 0.0210 (=)	0.2019 $\pm$ 0.0214 (+)	0.1277 $\pm$ 0.0394 (+)
Uniform random	0.7152 $\pm$ 0.0417 (+)	0.2630 $\pm$ 0.0921 (+)	0.1875 $\pm$ 0.0516 (+)	0.1291 $\pm$ 0.0125 (+)

Bold values indicate the best mean HV among the compared strategy-selection mechanisms within each case. The symbols in parentheses compare each alternative method with SW-ASS using the paired Wilcoxon signed-rank test. The symbols + and = indicate that SW-ASS is significantly better than or not significantly different from the corresponding method at $\alpha = 0.05$, respectively.

As shown in Table 3, SW-ASS achieves the highest mean HV in three out of the four representative complex scenarios, namely Cases 5, 10, and 11. In Case 9, cumulative reward selection obtains a slightly higher mean HV, but the Wilcoxon result indicates that the difference from SW-ASS is not statistically significant. Compared with uniform random selection, SW-ASS avoids unguided strategy switching and provides more stable search behavior. Compared with cumulative reward selection, SW-ASS relies on recent strategy performance rather than long-term accumulated feedback only, thereby reducing the influence of outdated rewards. These results indicate that the sliding-window adaptive selection mechanism contributes positively to the performance and stability of SW-MSACO in complex routing scenarios.

5.4. Parameter Sensitivity Analysis

5.4.1. Sensitivity to Objective Weights

To examine whether the conclusions of SW-MSACO remain valid under different operational preferences, an additional weight-sensitivity experiment was conducted on four representative complex scenarios, including Cases 5, 9, 10, and 11. Five objective-weight configurations were tested: balanced preference ($w_1, w_2 = (1, 1)$), baseline safety-priority preference (1, 2), timeliness-priority preference (2, 1), strong safety-priority preference (1, 3), and strong timeliness-priority preference (3, 1). Each configuration was independently executed twenty times.

As shown in Table 4, SW-MSACO maintains relatively stable HV values under different weight configurations. Although the best-performing weight pair varies across scenarios, no setting leads to severe performance degradation. This indicates that the conclusions of SW-MSACO remain generally valid under balanced, safety-priority, and timeliness-priority settings. The baseline setting ($w_1, w_2 = (1, 2)$) achieves competitive HV values across the representative scenarios and is therefore adopted as the safety-priority configuration in the main experiments.

Table 4. Weight sensitivity results under different objective-weight settings.

(w_1, w_2)	Mean HV $\pm$ Std. Dev.			
	Case 5	Case 9	Case 10	Case 11
(1, 1)	0.7506 $\pm$ 0.1759	0.2357 $\pm$ 0.0884	0.1887 $\pm$ 0.0403	0.1792 $\pm$ 0.0532
(1, 2)	0.7843 $\pm$ 0.1917	0.2391 $\pm$ 0.0864	0.1956 $\pm$ 0.0368	0.1751 $\pm$ 0.0547
(2, 1)	0.7426 $\pm$ 0.1909	0.2493 $\pm$ 0.0898	0.1967 $\pm$ 0.0428	0.1806 $\pm$ 0.0537
(1, 3)	0.7744 $\pm$ 0.1523	0.2394 $\pm$ 0.0846	0.1820 $\pm$ 0.0460	0.1755 $\pm$ 0.0547
(3, 1)	0.7850 $\pm$ 0.1847	0.2478 $\pm$ 0.0875	0.1998 $\pm$ 0.0380	0.1836 $\pm$ 0.0578

5.4.2. Sensitivity to Sliding-Window Capacity

To examine the sensitivity of the proposed algorithm to a key tuning parameter, experiments are conducted under a fixed setting of 80 nodes and 100 orders to evaluate the effect of sliding-window capacity on optimization performance. Detailed quantitative results obtained over 20 independent runs are summarized in Table 5.

Table 5. Sensitivity analysis under different sliding window capacities.

Window Capacity	Mean HV	Std. Dev.	Average Time (s)
5	0.2314	0.0412	110.12
10	0.3089	0.0173	120.91
15	0.2710	0.0210	125.45
20	0.2584	0.0198	134.80

As shown in Table 5, when the window capacity is too small ($W_{cap} = 5$), the average runtime slightly decreases to 110.12 s. However, the mean HV deteriorates to 0.2314, accompanied by a relatively high fluctuation level (Std. Dev. = 0.0412). This performance degradation is mainly attributed to the stochastic sensitivity of the short-term adaptive memory queue. A limited window capacity may overemphasize transient local improvements, thereby excessively increasing the selection probability of exploitation-oriented strategies such as S1. Consequently, the ant colony may converge prematurely toward narrow suboptimal regions in the solution space, reducing the search diversity required to construct a well-distributed Pareto frontier.

In contrast, when the window capacity is excessively large ($W_{cap} = 20$), performance also deteriorates, with the mean HV decreasing to 0.2584 while the average runtime increases to 134.80 s. An overly large memory window retains outdated feedback information from earlier generations, introducing excessive historical inertia into the adaptive selection process. Consequently, the roulette-based selection mechanism becomes overly dependent on historical information, resulting in delayed adaptation and reduced search efficiency during the metaheuristic optimization process.

Among the tested configurations, the configuration with $W_{cap} = 10$ achieves a balanced trade-off between exploration capability and exploitation stability. It attains the highest mean HV (0.3089) together with a relatively low variance (Std. Dev. = 0.0173), while maintaining acceptable computational overhead (120.91 s). These results indicate that $W_{cap} = 10$ effectively reduces stochastic sampling noise while preserving adaptive responsiveness in multi-UAV path coordination under high-workload conditions.

5.5. Discussion of Limitations

Despite the demonstrated optimization stability and constraint compliance under complex task-load conditions, several limitations of the proposed SW-MSACO framework should be acknowledged.

First, the current study adopts a reduced-order energy-cost model for simulation-based feasibility evaluation. This model captures the effects of travel distance, remaining payload, and cruising speed on route-level energy cost, allowing all compared algorithms to be evaluated under the same battery constraint. However, the model should not be interpreted as a fully calibrated aerodynamic power model for a specific UAV platform. The adopted parameters are normalized simulation coefficients rather than experimentally identified physical constants. Therefore, the reported results mainly demonstrate the algorithmic behavior of SW-MSACO under controlled energy constraints, rather than providing direct experimental validation of UAV energy consumption in real flight operations.

Second, the current simulation assumes that the planned trajectories are executed as designed. In real low-altitude urban environments, UAV trajectories may deviate from planned routes due to GNSS degradation, positioning uncertainty, wind disturbance, collision-avoidance maneuvers, communication delay, dynamic deconfliction procedures, temporary no-fly zones, holding patterns, and airspace-management waiting time [37,38]. These factors may increase actual flight duration and energy expenditure compared with the nominal route-level values estimated in this study.

Third, the current safety-risk matrix is generated within the simulation environment and is used to evaluate the relative safety exposure of different route segments. Although this setting is sufficient for comparing the search behavior of different algorithms under consistent risk constraints, it does not fully represent real urban airspace risk distributions. Practical UAV deployment would require GIS-based information, population-density data, building-height distributions, no-fly-zone boundaries, and dynamic airspace-management constraints. Accordingly, the proposed risk representation should be regarded as a controlled simulation abstraction rather than a complete operational risk model.

Another limitation is that the current comparative study focuses mainly on fixed-strategy ACO variants and a static hybrid ACO baseline. This design was adopted to isolate the contribution of the proposed multi-strategy pheromone evolution pool and the SW-ASS mechanism under the same solution-construction and simulation-evaluation framework. However, broader comparisons with general-purpose multi-objective optimizers, such as NSGA-II, MOEA/D and SPEA2, are still required to further assess the relative performance of SW-MSACO across different optimization paradigms. This will be considered as an important direction for future work.

Therefore, the experimental results should be interpreted as simulation-based evidence under simplified and controlled operational conditions rather than direct validation of real-world urban UAV deployment. The proposed framework demonstrates promising optimization performance for the formulated CMURP, but its practical applicability still requires further validation under realistic urban airspace data and operational uncertainty. Future work will extend the current framework by incorporating high-fidelity energy calibration, stochastic wind-field disturbance, dynamic exclusion zones, emergency re-routing, battery degradation, and GIS-based urban airspace data to further evaluate the practical applicability of the proposed method.

6. Conclusions

This paper tackles the combinatorial explosion and premature convergence challenges of multi-objective drone routing optimization under tight constraints, which remain fundamental limitations of traditional ant colony optimization frameworks. To enhance adaptability and diversity maintenance in complex urban logistics, this study proposes a SW-MSACO algorithm that incorporates a multi-strategy pheromone evolution pool and a Sliding-Window Adaptive Strategy Selection (SW-ASS) mechanism.

The proposed SW-ASS dynamically adjusts operator selection based on short-term memory feedback, emphasizing recent search performance, while the pheromone evolution pool introduces diversified deposition behaviors to mitigate search stagnation and improve responsiveness in highly coupled multi-objective landscapes. In addition, the weighted fitness formulation provides a preference-guidance mechanism for fitness evaluation and pheromone updating under different operational preferences.

Extensive experiments on multi-scale routing scenarios and stress-test benchmarks demonstrate that SW-MSACO achieves competitive or superior performance compared with fixed and hybrid heuristics, particularly in large-scale networks and high-load scenarios. Sensitivity studies further verify the effectiveness of the sliding-window mechanism, with an intermediate window size achieving a favorable balance between stability and exploration.

In summary, SW-MSACO provides an effective enhancement to swarm-intelligence-based multi-objective optimization frameworks for urban drone logistics. Nevertheless, the current results should be interpreted within the simplified simulation assumptions adopted in this study, and further validation under realistic urban airspace conditions is still required. Future work will focus on investigating continuous indicator-based reward functions, such as HV-improvement-based or archive-quality-based feedback, as well as parallel acceleration, uncertainty-aware modeling of wind disturbances [39], and dynamic constraint handling in geo-fenced airspaces to further improve robustness in practical deployment scenarios.

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

Funding: This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 62406152 and 62502097, in part by the Natural Science Foundation of Tianjin under Grant 24JCQNJC02100, and in part by the College Students' Innovative Entrepreneurial Training Plan Program under Grant 202510055078.

Data Availability Statement: The data will be made available upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned aerial vehicle
CMURP	Complex multi-objective urban routing problem
ACO	Ant colony optimization
GA	Genetic algorithms
PSO	Particle swarm optimization
AOS	Adaptive operator selection
SW-MSACO	Sliding-window adaptive multi-strategy ant colony optimization
SW-ASS	Sliding-window adaptive strategy selection
VRP	Vehicle routing problems
FN	Feasible neighborhood
CD	Crowding distance
HV	Hypervolume
Std. Dev.	Standard deviation

References

1. Jazairy, A.; Persson, E.; Brho, M.; von Haartman, R.; Hilletoft, P. Drones in last-mile delivery: A systematic literature review from a logistics management perspective. *Int. J. Logist. Manag.* **2025**, *36*, 1–62. <https://doi.org/10.1108/IJLM-04-2023-0149>.
2. Oakey, A.; Grote, M.; Pilko, A.; Krol, J.; Blakesley, A.; Cherrett, T.; Martinez-Sykora, A.; Anvari, B. Comparative modelling of drone interventions with traditional logistics modes for expedited and equitable deliveries. *npj Sustain. Mobil. Transp.* **2025**, *2*, 37. <https://doi.org/10.1038/s44333-025-00050-8>.
3. Shuaibu, A.S.; Mahmoud, A.S.; Sheltami, T.R. A Review of Last-Mile Delivery Optimization: Strategies, Technologies, Drone Integration, and Future Trends. *Drones* **2025**, *9*, 158. <https://doi.org/10.3390/drones9030158>.
4. Dantzig, G.B.; Ramser, J.H. The Truck Dispatching Problem. *Manag. Sci.* **1959**, *6*, 80–91. <https://doi.org/10.1287/mnsc.6.1.80>.
5. Lu, Y.; Wen, W.; Igorevich, K.K.; Ren, P.; Zhang, H.; Duan, Y.; Zhu, H.; Zhang, P. UAV Ad Hoc Network Routing Algorithms in Space–Air–Ground Integrated Networks: Challenges and Directions. *Drones* **2023**, *7*, 448. <https://doi.org/10.3390/drones7070448>.
6. Javed, S.; Hassan, A.; Ahmad, R.; Ahmed, W.; Ahmed, R.; Saadat, A.; Guizani, M. State-of-the-Art and Future Research Challenges in UAV Swarms. *IEEE Internet of Things Journal* **2024**, *11*, 19023–19045. <https://doi.org/10.1109/JIOT.2024.3364230>.
7. Kumar, P.; Pal, K.; Govil, M.C. Comprehensive Review of Path Planning Techniques for Unmanned Aerial Vehicles (UAVs). *ACM Comput. Surv.* **2025**, *58*, 1–44. <https://doi.org/10.1145/3737280>.
8. Meng, W.; Zhang, X.; Zhou, L.; Guo, H.; Hu, X. Advances in UAV Path Planning: A Comprehensive Review of Methods, Challenges, and Future Directions. *Drones* **2025**, *9*, 376. <https://doi.org/10.3390/drones9050376>.
9. Jasim, A.N.; Fourati, L.C. Guided Genetic Algorithm for Solving Capacitated Vehicle Routing Problem with Unmanned-Aerial-Vehicles. *IEEE Access* **2024**, *12*, 106333–106358. <https://doi.org/10.1109/ACCESS.2024.3438079>.
10. Dorigo, M.; Maniezzo, V.; Colnori, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. B Cybern.* **1996**, *26*, 29–41. <https://doi.org/10.1109/3477.484436>.
11. Dorling, K.; Heinrichs, J.; Messier, G.G.; Magierowski, S. Vehicle Routing Problems for Drone Delivery. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 70–85. <https://doi.org/10.1109/TSMC.2016.2582745>.
12. Wang, X.; Choi, T.M.; Liu, H.; Yue, X. Novel Ant Colony Optimization Methods for Simplifying Solution Construction in Vehicle Routing Problems. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3132–3141. <https://doi.org/10.1109/TITS.2016.2542264>.
13. Frías, N.; Johnson, F.; Valle, C. Hybrid Algorithms for Energy Minimizing Vehicle Routing Problem: Integrating Clusterization and Ant Colony Optimization. *IEEE Access* **2023**, *11*, 77366–77382. <https://doi.org/10.1109/access.2023.3325787>.
14. Pei, J.; Mei, Y.; Liu, J.; Zhang, M.; Yao, X. Adaptive Operator Selection for Meta-Heuristics: A Survey. *IEEE Trans. Artif. Intell.* **2025**, *6*, 1991–2012. <https://doi.org/10.1109/TAI.2025.3545792>.
15. Fialho, Á.; Da Costa, L.; Schoenauer, M.; Sebag, M. Analyzing bandit-based adaptive operator selection mechanisms. *Ann. Math. Artif. Intell.* **2010**, *60*, 25–64. <https://doi.org/10.1007/s10472-010-9213-y>.
16. Rojas Viloria, D.; Solano-Charris, E.L.; Muñoz-Villamizar, A.; Montoya-Torres, J.R. Unmanned aerial vehicles/drones in vehicle routing problems: A literature review. *Int. Trans. Oper. Res.* **2021**, *28*, 1626–1657. <https://doi.org/10.1111/itor.12783>.
17. Guerriero, F.; Surace, R.; Loscrí, V.; Natalizio, E. A multi-objective approach for unmanned aerial vehicle routing problem with soft time windows constraints. *Appl. Math. Model.* **2014**, *38*, 3939–3952. <https://doi.org/10.1016/j.apm.2013.12.001>.
18. Zhao, H.; Gu, M.; Qiu, S.; Zhao, A.; Deng, W. Dynamic Path Planning for Space-Time Optimization Cooperative Tasks of Multiple Unmanned Aerial Vehicles in Uncertain Environment. *IEEE Transactions on Consumer Electronics* **2025**, *71*, 7673–7682. <https://doi.org/10.1109/TCE.2025.3593383>.
19. Ozkan, O. Multi-objective optimization of transporting blood products by routing UAVs: The case of Istanbul. *Int. Trans. Oper. Res.* **2023**, *30*, 3317–3345. <https://doi.org/10.1111/itor.13109>.
20. Thibbotuwawa, A.; Nielsen, P.; Banaszak, Z.; Bocewicz, G. Energy consumption in unmanned aerial vehicles: A review of energy-consumption models and their relation to the UAV routing. In *Information Systems Architecture and Technology*; Świątek, J., Borzemski, L., Wilimowska, Z., Eds.; Physica-Verlag: Berlin/Heidelberg, Germany, 2019; Volume 853, pp. 173–184. https://doi.org/10.1007/978-3-319-99996-8_16.
21. Thibbotuwawa, A.; Nielsen, P.; Banaszak, Z.; Bocewicz, G. Factors affecting energy consumption of unmanned aerial vehicles: An analysis of how energy consumption changes in relation to UAV routing. In *Information Systems Architecture and Technology*; Świątek, J., Borzemski, L., Wilimowska, Z., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 853, pp. 228–238. https://doi.org/10.1007/978-3-319-99996-8_21.
22. Kierzkowski, A.; Dziewoński, B.; Kaliszuk, K.; Kucharski, M. Evaluation of light electric flying-wing unmanned aerial system energy consumption during holding maneuver. *Energies* **2025**, *18*, 1300. <https://doi.org/10.3390/en18051300>.

23. Hamissi, A.; Dhraief, A. A Survey on the Unmanned Aircraft System Traffic Management. *ACM Comput. Surv.* **2023**, *56*, 1–37. <https://doi.org/10.1145/3617992>.
24. Braßel, H.; Zeh, T.; Lindner, M.; Fricke, H. Risk-Aware UAV Trajectory Optimization Using Open Urban GIS Data and Target Level of Safety Constraints. *Drones* **2025**, *9*, 666. <https://doi.org/10.3390/drones9100666>.
25. Kuo, R.J.; Edbert, E.; Zulvia, F.E.; Lu, S.H. Applying NSGA-II to vehicle routing problem with drones considering makespan and carbon emission. *Expert Syst. Appl.* **2023**, *221*, 119712. <https://doi.org/10.1016/j.eswa.2023.119712>.
26. DaCosta, L.; Fialho, A.; Schoenauer, M.; Sebag, M. Adaptive operator selection with dynamic multi-armed bandits. In Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO '08), Atlanta, GA, USA, 12–16 July 2008; pp. 913–920. <https://doi.org/10.1145/1389095.1389272>.
27. Deb, K. Multi-Objective Optimization Using Evolutionary Algorithms; John Wiley & Sons: Chichester, UK, 2001; pp. 1–497.
28. Stützle, T.; Hoos, H.H. MAX-MIN Ant System. *Future Gener. Comput. Syst.* **2000**, *16*, 889–914. [https://doi.org/10.1016/S0167-739X\(00\)00043-1](https://doi.org/10.1016/S0167-739X(00)00043-1).
29. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. <https://doi.org/10.1109/4235.996017>.
30. Aleti, A.; Moser, I. A Systematic Literature Review of Adaptive Parameter Control Methods for Evolutionary Algorithms. *ACM Comput. Surv.* **2016**, *49*, 1–35. <https://doi.org/10.1145/2996355>.
31. Zitzler, E.; Thiele, L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* **1999**, *3*, 257–271. <https://doi.org/10.1109/4235.797969>.
32. Zitzler, E.; Thiele, L.; Laumanns, M.; Fonseca, C.M.; Da Fonseca, V.G. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans. Evol. Comput.* **2003**, *7*, 117–132. <https://doi.org/10.1109/TEVC.2003.810758>.
33. Ishibuchi, H.; Imada, R.; Setoguchi, Y.; Nojima, Y. How to Specify a Reference Point in Hypervolume Calculation for Fair Performance Comparison. *Evol. Comput.* **2018**, *26*, 411–440. https://doi.org/10.1162/evco_a_00226.
34. Ishibuchi, H.; Imada, R.; Setoguchi, Y.; Nojima, Y. Reference point specification in hypervolume calculation for fair comparison and efficient search. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '17), Berlin, Germany, 15–19 July 2017; pp. 585–592. <https://doi.org/10.1145/3071178.3071264>.
35. Veček, N.; Črepinšek, M.; Mernik, M. On the influence of the number of algorithms, problems, and independent runs in the comparison of evolutionary algorithms. *Appl. Soft Comput.* **2017**, *54*, 23–45. <https://doi.org/10.1016/j.asoc.2017.01.011>.
36. Burke, E.K.; Gendreau, M.; Hyde, M.; Ochoa, G.; Ozcan, E.; Qu, R.; Vazquez-Rodriguez, J.A. Hyper-heuristics: A survey of the state of the art. *J. Oper. Res. Soc.* **2013**, *64*, 1695–1724. <https://doi.org/10.1057/jors.2013.71>.
37. Li, Q.; Zhang, W.; Zheng, B. Robust Route-Speed Optimization for UAV Inspection Missions Under Wind Uncertainty. *Math. Comput. Appl.* **2026**, *31*, 84. <https://doi.org/10.3390/mca31030084>.
38. Thibbotuwawa, A.; Bocewicz, G.; Radzki, G.; Nielsen, P.; Banaszak, Z. UAV Mission Planning Resistant to Weather Uncertainty. *Sensors* **2020**, *20*, 515. <https://doi.org/10.3390/s20020515>.
39. Pedemonte, M.; Nesmachnow, S.; Cancela, H. A survey on parallel ant colony optimization. *Appl. Soft Comput.* **2011**, *11*, 5181–5197. <https://doi.org/10.1016/j.asoc.2011.05.042>.

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.