Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Enhanced Gold Mining Optimization with Dual-Layer Information Sharing and Dynamic Control for Multi-Objective Scheduling of Heterogeneous UAV

Siyuan Wei

siyuanwei2026@outlook.com

Nanning Normal University https://orcid.org/0000-0002-5676-0733

zhongming lin

Nanning Normal University

Research Article

Keywords: Gold Mining Optimization Algorithm, Two-Tier Information-Sharing Framework, Dynamic Solution Update Mechanism, Adaptive Pareto Dominance Strategy, Heterogeneous Unmanned Aerial Vehicle Swarms, Collaborative Task Scheduling, Multi-Objective Optimization

Posted Date: August 6th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10595287/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Enhanced Gold Mining Optimization with Dual-Layer Information Sharing and Dynamic Control for Multi-Objective Scheduling of Heterogeneous UAV Swarms

Zhongming Lin [1] **Siyuan Wei** [2 *]

1 Zhongming Lin; PhD, Nanning Normal University; institutional email: linzhongming@nnu.edu.cn

2 Siyuan Wei; PhD, Nanning Normal University; institutional email:siyuanwei@nnnu.edu.cn

* Correspondence:Siyuan Wei; e-mail: siyuanwei2026@outlook.com; Tel.:86+19807715400

Abstract：

Heterogeneous unmanned aerial vehicle (UAV) swarm cooperative scheduling constitutes a fundamental challenge in advancing UAV swarm intelligence. This problem entails the tightly coupled optimization of task allocation and sequencing, requiring simultaneous trade-offs among conflicting objectives, including mission completion time, energy consumption, and computational load balance, under multiple operational and physical constraints. To overcome critical limitations of the original Gold Mine Optimization Algorithm (GMO)- specifically, its insufficient population diversity and sluggish convergence toward the Pareto-optimal front- this paper proposes an Improved Gold Mine Optimization Algorithm with a dual-layer information-sharing framework and dynamic Pareto dominance (IGMO-DP). First, a dual-layer information-sharing mechanism is introduced that synergistically integrates local neighborhood enhancement with global elite guidance, thereby preserving population diversity while expediting convergence. Second, we design a dynamic exploration–exploitation balancing strategy based on adaptive step-size adjustment, enabling context-aware modulation of search behavior throughout the optimization process. Third, an elite archive maintenance scheme is incorporated, coupled with a dynamically updated Pareto dominance relationship to enhance both the convergence and uniformity of the final non-dominated solution set. Based on these algorithmic components, we formulate a multi-objective optimization model for heterogeneous UAV cooperative scheduling that jointly minimizes mission completion time, total energy expenditure, and inter-UAV load imbalance. The proposed IGMO-DP is rigorously evaluated across three representative complex operational scenarios. Comparative experiments demonstrate that IGMO-DP consistently surpasses the baselines, GMO, NSGA-II, MOPSO, and HMOEA in terms of convergence speed, diversity of the obtained solution set, and the quality of the Pareto front distribution. Quantitatively, the Inverted Generational Distance (IGD) metric improves by 14.2%–17.6%, while the Hypervolume (HV) metric increases by 4.9%–5.5%, thereby substantiating the algorithm's efficacy and competitive advantage for heterogeneous UAV swarm scheduling.

Keyword： Gold Mining Optimization Algorithm; Two-Tier Information-Sharing Framework; Dynamic Solution Update Mechanism; Adaptive Pareto Dominance Strategy; Heterogeneous Unmanned Aerial Vehicle Swarms; Collaborative Task Scheduling; Multi-Objective Optimization

1 Introduction

The deployment of clustered unmanned aerial vehicles (UAVs) for cooperative operations has solidified their status as a central research thrust in intelligent systems, underpinned by empirically validated performance across mission-critical domains, namely military reconnaissance, disaster response, environmental monitoring, and logistics distribution. Within the class of bio-inspired meta-heuristics, the ant colony optimization (ACO) algorithm (Dorigo & Stützle, 2004) has demonstrated consistent efficacy in UAV path planning, which is largely attributable to its biologically faithful emulation of pheromone-mediated collective navigation [1]. Likewise, particle swarm optimization (PSO) (Kennedy & Eberhart, 1995), grounded in the principles of swarm intelligence and social learning, has established a rigorous methodological foundation for UAV mission planning through its capacity for rapid convergence toward high-quality solutions in complex search spaces [2]. Extending this trajectory, other nature-inspired algorithms—including the whale optimization algorithm (WOA) (Mirjalili & Lewis, 2016) [3] and the grey wolf optimizer (GWO) (Mirjalili et al., 2014) [4]—have been rigorously adapted and empirically validated for UAV trajectory planning and task allocation, demonstrating substantial improvements in solution robustness and computational efficiency under realistic operational constraints.

Heterogeneous UAV formations—integrating functionally complementary platforms such as fixed-wing (long-endurance, high-speed), rotary-wing (vertical takeoff/landing, high maneuverability), and VTOL (Vertical Take-Off and Landing, combining hover capability with forward flight efficiency)systems—systematically exploit platform-specific strengths to achieve synergistic mission capability. This architectural diversity confers measurable advantages over homogeneous configurations in terms of operational adaptability, fault tolerance, and functional redundancy. In multi-objective evolutionary computation, seminal contributions include Zitzler and Thiele's (1999) strength Pareto evolutionary algorithm (SPEA), which pioneered archive-based elitism for preserving non-dominated solutions [5], and Zhang and Li's (2007) decomposition-based multi-objective evolutionary algorithm (MOEA/D), whose problem decomposition strategy markedly improves scalability and convergence stability in high-dimensional UAV scheduling problems [6]. However, heterogeneity fundamentally reshapes the optimization landscape: divergent dynamic models, heterogeneous energy consumption dynamics, asymmetric sensing/actuation capabilities, and variable communication bandwidths collectively impose stringent coupling requirements between task allocation and trajectory planning. Consequently, effective scheduling must simultaneously optimize multiple, inherently competing objectives—including mission time-to-completion, total energy expenditure, collision/risk probability, and spatial coverage fidelity—which renders the formulation a canonical multi-objective combinatorial optimization problem with tightly interdependent decision variables [7, 8].

Meta-heuristic algorithms are widely adopted for UAV cooperative scheduling because they provide principled, scalable strategies for navigating high-dimensional, non-convex, and constraint-dense solution spaces-scenarios that render exact methods computationally intractable. Deb et al.'s (2002) NSGA-II remains the most frequently cited benchmark in this domain, owing not only to its fast non-dominated sorting and crowding distance estimation but also to its proven ability to maintain solution-set diversity without sacrificing convergence speed [9]. Coello Coello et al. (2004) extended the PSO paradigm to multi-objective settings via multi-objective particle swarm optimization (MOPSO), introducing external archives and leader selection heuristics that enable stable approximation of the Pareto front [10]. More recently, Meng et al. (2025) advanced the state of the art by introducing an evolutionary multi-task optimization framework for multi-UAV path planning, wherein adaptive operator selection and cross-task knowledge transfer

demonstrably accelerate convergence while preserving solution diversity across heterogeneous mission scenarios [11].

The Gold Mining Optimization (GMO) algorithm is a novel entrant in the metaheuristic landscape. Zhang et al. (2025) conducted a systematic empirical evaluation of the original GMO, confirming its strong global exploration capability—rooted in three coordinated behavioral operators: prospecting (local exploitation), migration (global exploration), and collaborative excavation (information sharing)—particularly on multimodal benchmark functions [12]. However, when applied to heterogeneous UAV scheduling, the algorithm exhibits three critical deficiencies: (i) limited information diffusion due to the absence of explicit neighborhood topology or archive interaction; (ii) rigid position-updating rules insensitive to objective gradient or constraint violation severity; and (iii) lack of dedicated diversity preservation mechanisms, resulting in sparse and clustered Pareto fronts under conflicting objectives. For constrained optimization, Wang et al. (2008) proposed an adaptive penalty-based trade-off model that dynamically adjusts penalty weights based on a history of constraint violation, thereby ensuring feasibility progression without compromising objective improvement [13]. Rahnamayan et al. (2008) enhanced differential evolution (DE) with opposition-based learning to improve initial population quality and convergence reliability [14], while Das and Suganthan (2011) delivered a comprehensive taxonomy of DE variants and their convergence properties—both serving as essential references for algorithmic hybridization [15]. Critically, Li et al. (2025) introduced a constrained multi-objective optimization framework explicitly designed for dynamic UAV group coordination under partial interference awareness; experimental validation confirmed its superior tracking accuracy and decision stability in time-varying threat environments [16]. Concurrently, Rashmi et al. (2025) provided a methodologically rigorous review of PSO's theoretical evolution, parameter adaptation paradigms (e.g., inertia weight tuning, topology design), and verified engineering deployments—thereby establishing actionable design principles for UAV-specific scheduler development [17].

Despite these methodological advances, three fundamental limitations persist across the current literature on heterogeneous UAV collaborative scheduling: (a) Structural decoupling of task allocation and sequencing—whether sequential or loosely coordinated—inevitably sacrifices global optimality, as local subproblem optima rarely compose a globally optimal joint solution; (b) Static or weakly adaptive algorithmic designs fail to respond effectively to real-time environmental changes (e.g., sudden obstacle emergence, communication dropouts, or shifting mission priorities), undermining operational resilience; and (c) Suboptimal Pareto front uniformity—manifesting as gaps, clusters, or boundary bias—directly constrains decision-makers' capacity to identify well-distributed, practically implementable trade-off solutions. While other prominent meta-heuristics—including the cuckoo search algorithm (Yang & Deb, 2009) [18], biogeography-based optimization (Simon, 2008) [19], artificial bee colony algorithm (Karaboga & Basturk, 2007) [20], gravitational search algorithm (Rashedi et al., 2009) [21], and krill herd algorithm (Gandomi & Alavi, 2012) [22]—have achieved notable success in generic optimization benchmarks, their application to heterogeneous UAV scheduling remains largely heuristic: lacking formal integration of UAV-specific kinematic constraints, heterogeneous resource models, or mission-level objective hierarchies.

Recent methodological innovations reflect growing recognition of these challenges. Zhao et al. (2026) proposed a multi-stage task allocation strategy that tightly couples an improved pigeon-inspired optimization algorithm with a parallel auction-based contract network, enabling provably convergent real-time task redistribution under dynamic workload fluctuations [23]. Zheng et al. (2026) developed a two-layer cross-domain optimization framework in which the upper layer performs strategic task assignment, while the lower layer solves trajectory generation as an optimal control problem, ensuring strict adherence to UAV dynamical feasibility and temporal consistency [24]. Zerrougui et al. (2025) introduced a Pareto-enhanced PSO

variant for edge-aware UAV deployment, incorporating an adaptive, bounded-size external archive and replacement rules based on dominance and diversity metrics to sustain solution-set richness [25]. Xu et al. (2025) designed a dual-balance multi-objective evolutionary algorithm featuring both convergence acceleration (via reference-point-guided selection) and diversity maintenance (via niche-based crowding), achieving statistically significant improvements—+4.2% in inverted generational distance (IGD) and +2.3% in hypervolume (HV)—over five state-of-the-art baselines [26]. Liu et al. (2024) formulated a multi-objective variable neighborhood search (MOVNS) algorithm with problem-specific neighborhood structures (e.g., task swap, UAV reassignment, path perturbation) and adaptive shaking intensity, yielding superior Pareto front coverage in large-scale heterogeneous scheduling instances [27]. Wang et al. (2021) introduced a dynamic crowding distance–guided MOPSO variant, in which the crowding distance is recalculated online using local density estimation, thereby enhancing both convergence and spread in long-horizon coordination tasks [28].

Nevertheless, critical gaps remain unaddressed: (a) Zhao et al. (2026) and Zheng et al. (2026), though achieving tighter coupling, rely on centralized or hierarchical architectures whose computational complexity scales super linearly with fleet size—limiting deploy ability in large-scale (>50 UAV), rapidly evolving operational theaters [23, 24]; (b) As rigorously quantified by Li et al. (2025), the average recovery time following unanticipated disturbances (e.g., sensor failure or adversarial jamming) exceeds acceptable thresholds under incomplete interference awareness, indicating insufficient intrinsic adaptability rather than implementation inefficiency [29]; and (c) Liu et al. (2024) observed that the distribution uniformity of Pareto-optimal fronts in multi-objective optimization remains suboptimal and warrants further methodological advancement [27]. Similarly, Zerrougui et al. (2025) underscored the critical role of solution diversity preservation in large-scale heterogeneous systems, particularly when scaling to high-dimensional decision spaces [25].

2 Theoretical Foundations and Problem Statement

2.1 Formal Description of the Cooperative Scheduling Problem for Heterogeneous Unmanned Aerial Vehicle Swarms

The collaborative scheduling problem for heterogeneous unmanned aerial vehicle (UAV) groups can be formally defined as follows: Given a set of heterogeneous UAVs and a set of geographically distributed tasks, the objective is to determine (i) an assignment of tasks to UAVs and (ii) an execution order (sequence) for each UAV—such that multiple operational constraints are satisfied and several performance objectives are jointly optimized.

Let the UAV fleet be denoted by $U = \{U_1, U_2, \ldots, U_n\}$, where n represents the total number of UAVs. Each UAV U_i is characterized by a heterogeneous attribute vector $\mathcal{A}_i = \{v_i^{max}, E_i^{max}, l_i, c_i\}$, corresponding respectively to its maximum airspeed, maximum onboard energy capacity, payload capacity, and unit-distance energy consumption coefficient. , and maneuverability type (e.g., fixed-wing, rotary-wing, hybrid-wing). The task set is $\mathcal{T} = \{T_1, T_2, \ldots, T_M\}$, where each task T_j is associated with a spatial location $p_j = (x_j, y_j)$, a time window $[t_j^{start}, t_j^{end}]$, and a demand quantity d_j.

The decision variables comprise:

a binary task assignment matrix $X = [x_{ij}]_{N\times M}$, where $x_{ij} \in \{0,1\}$ indicates whether UAV U_i is assigned to execute task T_j; subject to the one-to-one assignment constraint: $\sum_{i=1}^{N} x_{ij} = 1, \forall j \in [1, M]$; for each UAV U_i, an ordered task sequence $\pi_i = (T_{\pi_i(1)}, T_{\pi_i(2)}, \ldots, T_{\pi_i(K_i)})$, representing the sequence in which the k_i assigned tasks are executed.

Trajectory evaluation: Given a sequence πi and UAV parameters, the flight path between consecutive task locations is computed using the Dubins path model [29], which respects the minimum turning radius and

other kinematic constraints of each UAV type. The flight time and energy consumption are then determined from the path length and UAV-specific speed/energy profiles. Thus, the optimization variables are only assignment and sequencing, while trajectories are deterministically evaluated – a pragmatic choice that dramatically reduces problem dimensionality while still accounting for UAV dynamics.

The optimization framework pursues three primary objectives:

(1) Minimization of makespan: Defined as the maximum completion time across all UAVs, i.e., $f_1 = \max_{i\in[1,N]} T_i^{total}$, where T_i^{total} denotes the total time required for UAV U_i to traverse its assigned trajectory and complete all associated tasks (including both flight time and on-site task execution time).

(2) Minimization of total energy consumption: Expressed as $f_2 = \sum_{i=1}^{N} E_i^{total}$, Energy consumption comprises two primary components: flight-related energy consumption and mission-execution energy consumption. Flight-related energy consumption is determined by the UAV's flight path length, cruising speed, and aerodynamic configuration.

(3) Maximization of workload balance: Quantified via the standard deviation of individual UAV completion times: $f_3 = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(T_i^{total} - \bar{T})^2}$, Here, T denotes the average mission duration across all drones. A smaller value of T indicates a more balanced distribution of operational workload.

Key constraints include:

• Task assignment uniqueness (each task assigned to exactly one UAV);

• Energy feasibility: $(\sum_{j\in\pi_i} e_{ij} \le e_i^{max})$, where e_{ij} denotes the energy consumed by U_i in traveling to and executing T_j;

• Temporal feasibility: Arrival time at task T_j must satisfy $(t_{ij}^{arrive} \in [t_j^{start}, t_j^{end}])$;

• Dynamic feasibility: Trajectories must respect UAV-specific kinematic limits—including minimum turning radius, maximum climb/descent angle, and acceleration bounds.

Table 1: UAV Physical Parameters (c: unit-distance energy consumption coefficient)

UAV Type	v^{max} (m/s)	E^{max} (kJ)	c (kJ/m)	payload (kg)	turn radius (m)
Fixed-wing	40	5000	0.25	10	50
Rotary-wing	20	3000	0.45	5	15
Hybrid-wing	30	4000	0.35	8	30
VTOL	35	4500	0.38	12	25

2.2 The basic principle of the Gold Mining optimization algorithm

The Gold Mining Optimization Algorithm (GMO), proposed by Zolfi in 2023, is a nature-inspired meta heuristic algorithm that models the exploratory, migratory, and collaborative behaviors exhibited by gold

miners in real-world mining operations. By abstracting these behavioral patterns into computational operators, GMO effectively balances global exploration and local exploitation across the solution space.

In the standard GMO model, each prospector represents a candidate solution, and the position vector x_i encodes the corresponding decision variables of the optimization problem. The algorithm operates through three sequential and functionally distinct phases:

(1) Exploration stage: Gold miners conduct random searches in unknown areas and update their positions using the following formula: $x_i^{new} = x_i + r_1 \cdot (x_{rand} - x_i)$;

(2) Migration stage: Gold miners move towards the current optimal position and the update formula is: $x_i^{new} = x_i + r_2 \cdot (x_{best} - x_i)$;

(3) Collaboration stage: Gold miners collaborate in search through information exchange: $x_i^{new} = x_i + r_3 \cdot (x_j - x_i)$.

Here, Where $r_1, r_2, r_3 \sim U[0,1]$ represent uniformly distributed random numbers. ; x_{rand} represents a randomly selected position of a gold miner agent; x_{best} denotes the current global best solution; and x_j signifies the position of another randomly selected gold miner agent.

The standard Gold Mining Optimization (GMO) algorithm performs global search through iterative alternation among the aforementioned mechanisms. It is characterized by structural simplicity and a minimal parameter set. However, when applied to complex optimization problems—such as multi-objective, heterogeneous unmanned aerial vehicle (UAV) swarm scheduling—Yanhua Zhang et al. [12] identified three principal limitations:

(1) Limited information sharing granularity: The algorithm relies solely on pairwise information exchange between randomly selected individuals, without incorporating localized neighborhood-based interaction. This restricts its capacity to sustain population diversity throughout the evolutionary process.

(2) Inflexible position update strategy: The step size remains fixed throughout the optimization process and lacks adaptive adjustment in response to convergence behavior, thereby increasing susceptibility to premature convergence and stagnation in local optima during later iterations.

(3) Inadequate multi-objective support: As originally formulated for single-objective optimization, the standard GMO lacks dedicated mechanisms—such as Pareto dominance handling, objective normalization, or archive maintenance—to effectively resolve conflicting objectives or approximate the Pareto-optimal front.

2.3 Research motivation

Building upon the foregoing analysis, the primary motivation of this study is to enhance the Gold Mining Optimization (GMO) algorithm by refining its information sharing and update mechanisms, thereby improving its efficacy in addressing multi-objective, heterogeneous unmanned aerial vehicle (UAV) swarm collaborative scheduling problems. To achieve this objective, three critical research questions are systematically addressed: (1) How can an information sharing strategy be designed to simultaneously support local fine-grained search and global exploratory guidance? (2) How can the search step size be adaptively adjusted to balance exploration and exploitation across diverse problem landscapes effectively? (3) How can the Pareto-optimal solution set be maintained with both high convergence and uniform distribution within the multi-objective optimization framework?

3 Improved Gold Mining Optimization Algorithm (IGMO-DP)

3.1 Overall architecture of the algorithm

This study proposes an enhanced variant of the Gold Mining Optimization (GMO) algorithm, termed the Improved Gold Mining Optimization with Dual-layer Information Sharing and Dynamic Pareto Dominance (IGMO-DP). IGMO-DP systematically extends the original GMO framework through three key enhancements: (1) a dual-layer information sharing mechanism that facilitates both infra- and inter-population knowledge transfer; (2) a dynamic position update strategy informed by adaptive neighborhood exploration; and (3) a refined multi-objective optimization scheme integrating dynamic Pareto dominance adjustment for improved convergence and diversity preservation. As illustrated in Figure 1, the overall architecture comprises seven core components: population initialization, fitness evaluation, dual-layer information sharing, dynamic position updating, elite archive maintenance, dynamic Pareto dominance updating, and termination criterion assessment.

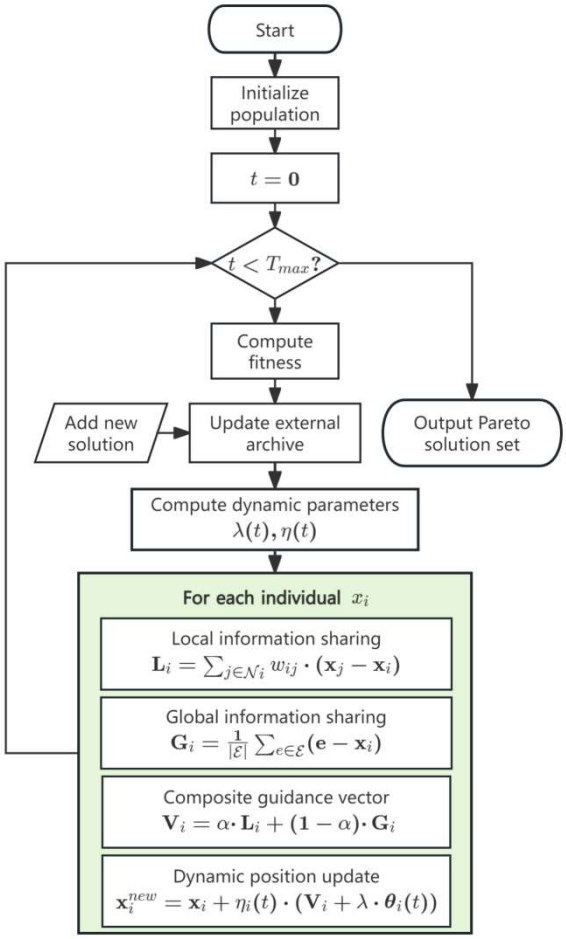


Figure 1: Flowchart of the Improved Gold Mining Optimization Algorithm (IGMO-DP)

Algorithm 1 IGMO-DP Main Framework

1. Initialize the population$P = x_1, x_2, \dots, x_N$, and adopt the chaotic mapping strategy to enhance the initial diversity.
2. Initialize the external archive $\mathcal{A} = \emptyset$.
3. Initialize the iteration count t = 0.
4. While $t < T_{max}$ do
5. Calculate the multi-objective fitness value F(xi) for each individual in the population $F(x_i) = \{f_1(x_i), f_2(x_i), \dots, f_p(x_i)\}$
6. Update external archive A (for maintaining non-dominated solution set, using the strategy in Section 3.4.1)
7. Calculate dynamic parameters based on the current iteration number t and population status: equilibrium coefficient $\alpha(t)$, step size η
8. For each individual x_i in P do
9. Share local information: Calculate the local direction vector Li according to formula (1)
10. Share global information: Calculate the global direction vector Gi according to formula (2)
11. Combine direction vectors: $V_i = (1 - \alpha(t))L_i + \alpha(t)G_i$
12. Update dynamic position: Update the position of xi according to formula (5)
13. End For
14. $t = t + 1$
15. End While
16. Return $\mathcal{A}$

3.2 Two-layer information sharing mechanism

In response to the challenges identified in Section 2.3—specifically, Challenge ① and the inherent limitation of the existing GMO information sharing mechanism, which relies solely on a single, uniform paradigm—this study introduces a dual-layer information sharing strategy. This strategy integrates local neighborhood enhancement with global elite guidance to preserve population diversity and improve convergence efficiency simultaneously.

3.2.1 Local neighborhood enhancement sharing

Local information sharing achieves fine-grained information exchange among individuals through the topological neighborhood structure. The neighborhood set $\mathcal{N}_i$ of an individual x_i is defined as the set consisting of the K_{neigh} nearest individuals in terms of Euclidean distance to this individual. The calculation method of the local guiding vector is as follows:

$$L_i = \sum_{j \in \mathcal{N}_i} w_{ij} \cdot (x_j - x_i) \quad (1)$$

Here, $|\mathcal{N}_i| = K_{neigh}$, and w_{ij} represents the weight coefficient based on fitness sorting, and adopts the weighted average fusion strategy:$w_{ij} = \frac{\frac{1}{rank(x_j)}}{\sum_{k \in \mathcal{N}_i} \frac{1}{rank(x_k)}}$, $rank(x_j)$ represents the non-dominated ranking level of the individual x_j in the current population. Individuals located on the Pareto front have rank = 1, those dominated by the front have rank = 2, and so on. This design assigns greater weights to individuals with higher fitness in the neighborhood, guiding the search toward the higher-quality area. The proposed local information-sharing mechanism offers two key advantages: first, it preserves local population diversity

through fine-grained neighborhood-level interaction, mitigating the risk of premature convergence; second, by integrating fitness-ranked information via a normalized weighting scheme, it enhances the directional and efficiency of the evolutionary search.

3.2.2 Global Elite-led Sharing

Global information sharing is guided by the elite solution set, which directs the overall search direction of the population. The elite solution set $\mathcal{E}$ is defined as the N_e individuals (in this paper, $N_e = \min(0.2 \cdot |\mathcal{A}|, 20)$) that rank higher in the non-dominated sorting in the external archive. The calculation method of the global guiding vector is as follows:

$$G_i = \frac{1}{|\mathcal{E}|}\sum_{e\in\mathcal{E}}(e - x_i) \quad (2)$$

The global information sharing mechanism offers the following advantages: The elite solution set constitutes the current best approximation of the Pareto-optimal frontier, derived from ongoing search efforts. By leveraging this set as a reference, the population is effectively guided toward the true Pareto frontier, thereby enhancing the algorithm's convergence rate.

3.2.3 Adaptive information sharing strategy fusion

To achieve a balance between local exploration and global development, an adaptive fusion coefficient $\alpha(t)$ is designed, which dynamically adjusts the information sharing strategy based on the iterative process:

$$\alpha(t) = \alpha_{min} + (\alpha_{max} - \alpha_{min}) \cdot e^{-\beta\cdot(t/T_{max})} \quad (3)$$

Here, α_{max} and α_{min} represent the upper and lower limits of the fusion coefficient, with $\alpha_{max} = 0.8$ used to ensure the initial intensity of the global search, and $\alpha_{min} = 0.2$ used to guarantee the local refinement capability in the later stage. The value range is determined on a simplified version of Scenario S2). β is the attenuation factor (currently set as $\beta = 5$), t is the current iteration number, and T_{max} is the maximum number of iterations. In the early stage of the algorithm, the value of α is larger, focusing on the sharing of global information to quickly locate high-quality areas; in the later stage of the algorithm, the value of α is smaller, focusing on sharing local information to conduct refined searches.

The comprehensive reference vector is expressed as:

$$V_i = (1 - \alpha(t)) \cdot L_i + \alpha(t) \cdot G_i \quad (4)$$

Thus, in the early stage ($\alpha \approx 0.2$), the global information G_i dominates, promoting global exploration; in the later stage ($\alpha \approx 0.8$), local information L_i dominates, facilitating fine-tuning. This aligns with the textual description.

3.3 Dynamic information update mechanism

In response to Issue ② identified in Section 2.3 and the inflexibility inherent in the conventional GMO update mechanism, this study proposes a dynamic update mechanism grounded in adaptive step-size adjustment. The proposed mechanism enables real-time modulation of the search step size according to the prevailing search landscape, thereby enhancing algorithmic responsiveness and convergence efficiency.

3.3.1 Adaptive step size adjustment

The step size factor $\eta_i(t)$ is adaptively adjusted according to the evolutionary state of the individual's fitness, and the specific formula is as follows:

$$\eta_i(t) = \eta_0 \cdot \left(1 + \gamma \cdot \frac{\Delta f_i(t)}{\max_j \Delta f_j(t)}\right) \cdot \left(1 - \frac{t}{T_{max}}\right)^{\delta} \quad (5)$$

Here, note that x_i^t and x_i^{t-1} denote the positions of individual i in the current and previous iterations, respectively. Therefore, the algorithm must maintain a memory of each individual's previous position. η_0 represents the initial step size, and$\Delta f_i(t) = \sum_{k=1}^{p} |f_k(x_i^t) - f_k(x_i^{t-1})|$ represents the fitness improvement of an individual x_i in the current iteration (that is, the sum of the absolute values of the changes in each objective function value). γ is the adjustment coefficient, and δ is the decay index. γ = 0.5 and δ = 0.8. These parameters are determined through parameter sensitivity analysis. This design endows the step size with dual adaptive capabilities: in the early stage of the search, the step size is set to a larger value, to facilitate global exploration; in the later stage of the search, the step size gradually decreases, which is beneficial for local refinement. At the same time, individuals with significant fitness improvement can obtain a larger step size, thereby accelerating the search for high-quality regions.

3.3.2 Directional adaptive adjustment

In addition to adapting the step length, the search direction must be dynamically adjusted to local terrain characteristics. To this end, a gradient-based direction adjustment factor is introduced, formulated as follows:

$$\theta_i(t) = \frac{\nabla f(x_i)}{\|\nabla f(x_i)\| + \epsilon} \quad (6)$$

Here, $\nabla f(x_i)$represents the gradient estimate of the objective function at x_i, which is calculated using the central difference method as $\nabla f_k(x_i) \approx (f_k(x_i + \Delta x) - f_k(x_i - \Delta x))/(2\Delta x)$; $\epsilon = 10^{-8}$ is a small constant used to avoid division by zero situations. In the multi-objective case, the gradient estimates need to be calculated separately for each objective and then weighted by the current non-dominated level.

It should be noted that in multi-objective settings, the Pareto front may exhibit non-convex or discontinuous geometries. Therefore, the gradient estimate in Eq. (6) serves as a heuristic local search direction rather than a mathematically exact gradient, which is a common practice in meta-heuristic algorithms for multi-objective optimization.

Taking into account both the step size and direction adjustment, the position update formula is as follows:

$$x_i^{new} = x_i + \eta_i(t) \cdot (V_i + \lambda \cdot \theta_i(t)) \quad (7)$$

λ represents the direction adjustment weight, with a value of 0.3, to balance the information guidance (70%) and gradient descent (30%).

3.4 Elite retention and archive maintenance

To address the issue ③ raised in Section 2.3 and to retain the high-quality solutions discovered during the algorithm's iterative process, the elite retention strategy and the external archive maintenance mechanism are introduced.

3.4.1 External file maintenance

External Archive $\mathcal{A}$ maintains the set of non-dominated solutions identified throughout the algorithm's execution. Its archive management strategy comprises two core operations:

(1) Insertion: Upon generating a new solution x_{new}, its dominance relationship with all existing archive members is evaluated. Specifically:

•If x_{new} is dominated by at least one solution in the archive, it is discarded;

•If x_{new} dominates one or more archive members; those dominated solutions are removed prior to inserting x_{new};

•If x_{new} is mutually non-dominated with all current archive members, it is directly inserted.

(2) Culling: When the archive size exceeds the predefined capacity limit A_{max}, a crowding-distance-based pruning mechanism is applied—specifically, the solution associated with the smallest crowding distance is removed—to preserve diversity and ensure uniform coverage of the approximated Pareto front [9].

3.4.2 Dynamic Pareto dominance

To enhance the algorithm's selection pressure during convergence, a dynamic Pareto dominance relationship is introduced. The dynamic dominance threshold $\tau(t)$ decreases with the iterative process. In the initial stage, slightly inferior solutions are allowed to be retained to maintain diversity, while in the later stage, strict screening is carried out to accelerate convergence:

$$\tau(t) = \tau_0 \cdot \left(1 - \frac{t}{T_{max}}\right) \quad (8)$$

$\tau_0$is set to 0.1 to ensure a 10% target slack in the initial stage This value was determined through a preliminary experiment.

Individual x_a dynamically dominates individual x_b (denoted as $x_a \prec_\tau x_b$) if and only when:

$$\forall k \in \{1, \ldots, p\}: f_k(x_a) \leq f_k(x_b) - \tau(t) \cdot (f_k^{max} - f_k^{min}) \quad (9)$$

And at least one strict inequality holds. Here, f_k^{max} and f_k^{min} represent the maximum and minimum values of the k-th objective in the current population, and they are updated once per generation based on the maximum and minimum values of each objective in the current population.

The advantage of dynamic Pareto dominance lies in the fact that, in the early stage of the algorithm, the dominance threshold is relatively large, and the dominance conditions are relaxed while maintaining the diversity of the population; in the later stage of the algorithm, the threshold gradually decreases, restoring the standard Pareto dominance and enhancing the convergence pressure.

3.5 Analysis of algorithm complexity

Let the population size be N, the number of objective functions be p, the maximum number of iterations be T_{max}, and the upper limit of the size of the external archive be A_{max}. The main computational costs of each round of iteration for IGMO-DP include:

(1) Fitness evaluation: $O(N \cdot p \cdot L)$, where L is the complexity of the target function;

(2) Neighborhood construction: Accelerated using KD tree, with complexity $O(N \log N)$);

(3) Calculation of local guiding vector: $O(N \cdot K)$, where K is the neighborhood size;

(4) Calculation of global guiding vector: $O(N \cdot A_{max})$;

(5) Maintenance of external archive: Worst case $O(A_{max}^2 \cdot p)$.

The overall time complexity is $O(T_{max} \cdot (N \cdot p \cdot L + N \log N + N \cdot A_{max} + A_{max}^2 \cdot p))$. Under typical parameter settings（$N = 100$, $A_{max} = 100$, $T_{max} = 200$）, this complexity is comparable to that of mainstream multi-objective evolutionary algorithms (such as NSGA-II), and is feasible for practical applications. The runtime of IGMO-DP is approximately 2-3 minutes (MATLAB R2020a, Intel Core i7-10700K), meeting the requirements of engineering applications.

4 Heterogeneous Unmanned Aerial Vehicle Group Cooperative Scheduling Multi-Objective Optimization Model

4.1 Formal Definition of the Problem

Based on the problem description in Section 2.1, this section develops a mathematical model for the multi-objective optimization of the collaborative scheduling of heterogeneous unmanned aerial vehicle groups.

Decision variable:

Task Allocation Matrix $X = [x_{ij}]_{N \times M}$, $x_{ij} \in 0,1$;

The task execution sequence vector $\Pi = (\pi_1, \pi_2, \ldots, \pi_N)$, where $\pi_i$represents the task sequence of the unmanned aerial vehicle U_i.

Objective function:

（1）Minimization of task completion time:

$$f_1(X,\Pi)=\max_{i\in[1,N]}\left(\sum_{k=1}^{K_i} t_{i,\pi_i(k)}^{fly}+\sum_{k=1}^{K_i} t_{i,\pi_i(k)}^{exec}\right) \quad (10)$$

Among them, $t_{i,\pi_i(k)}^{fly}$ represents the time required for the unmanned aerial vehicle U_i to fly from its current position (or the previous task position) to the task $T_{\pi_i(k)}$, and it is calculated using the Dubins path model to take into account the dynamic constraints of the unmanned aerial vehicle; $t_{i,\pi_i(k)}^{exec}$ represents the task execution time.

（2）Minimization of energy consumption:

$$f_2(X,\Pi)=\sum_{i=1}^{N}\left(\sum_{k=1}^{K_i} e_{i,\pi_i(k)}^{fly}+\sum_{k=1}^{K_i} e_{i,\pi_i(k)}^{exec}\right) \quad (11)$$

Among them, $e_{i,\pi_i(k)}^{fly}$ represents the flight energy consumption, which is related to the path length and the type of the drone; $e_{i,\pi_i(k)}^{exec}$ represents the energy consumption for task execution.

（3）Load balancing maximization (equivalent to minimizing load variance):

$$f_3(X,\Pi)=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{k=1}^{K_i} t_{i,\pi_i(k)}^{total}-\bar{T}\right)^2} \quad (12)$$

$\sum_{k=1}^{K_i} t_{i,\pi_i(k)}^{total}$ represents the total mission time of the unmanned aerial vehicle U_i. $\bar{T}=\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K_i} t_{i,\pi_i(k)}^{total}$ represents the average mission time of all unmanned aerial vehicles.

Constraints:

Uniqueness of Task Allocation:

$$\sum_{i=1}^{N} x_{ij}=1,\quad \forall j\in[1,M] \quad (13)$$

Energy constraint:

$$\sum_{k=1}^{K_i} e_{i,\pi_i(k)}^{total}\le E_i^{max},\quad \forall i\in[1,N] \quad (14)$$

$e_{i,\pi_i(k)}^{total}=e_{i,\pi_i(k)}^{fly}+e_{i,\pi_i(k)}^{exec}$ represents the total energy consumption, comprising both flight-related and mission-specific energy expenditure.

Time window constraint:

$$t_{ij}^{arrive}\in[t_j^{start},t_j^{end}],\quad \forall i,j \text{ with } x_{ij}=1 \quad (15)$$

Mechanical feasibility constraints: The flight path must comply with dynamic constraints, including the unmanned aerial vehicle's minimum turning radius and maximum climbing angle.

4.2 Multi-Objective Optimization Solution Framework

The above multi-objective optimization model is integrated with the IGMO-DP algorithm to solve the collaborative scheduling problem of heterogeneous unmanned aerial vehicle groups. The specific process is as follows:

Step 1: Coding scheme design

A task sequence-based coding method is adopted. Each individual consists of two parts: the task allocation vector $a\in[1,N]^M$, which represents the drone number assigned to each task, and the task sequence vector $o\in\mathbb{R}^M$, which indicates the execution priority of the tasks. During the decoding process, the tasks

assigned to each drone are sorted according to their designated priority, thereby obtaining the task sequence π_i. This coding method ensures that the uniqueness constraint for task allocation is automatically satisfied.

Step 2: Constraint Handling

To address the energy constraint (14) and the time window constraint (15), a penalty function approach is employed: constraint violations are quantified and incorporated into the objective function as additive penalty terms. For the dynamic feasibility constraint, path generation explicitly incorporates Dublin's vehicle model to guarantee kinematic feasibility of all computed trajectories.

Specifically, for a given solution (X, Π), the total constraint violation is defined as:

$$CV(X,\Pi) = \sum_{i=1}^{N} \max(0, \sum_{j\in\pi_i} e_{ij}^{total} - E_i^{max}) + \sum_{i=1}^{N} \sum_{j\in\pi_i} \max(0, t_{ij}^{start} - t_{ij}^{arrive}, t_{ij}^{arrive} - t_{ij}^{end}) \quad (16)$$

The penalized objective function is then computed as:

$f_k^{penalized} = f_k + \rho \cdot CV$ for $k = 1,2,3$, where $\rho = 10^6$ is a sufficiently large penalty coefficient.

Step 3: Algorithm Solution

Input the encoded population into the IGMO-DP algorithm and perform iterative optimization according to the process described in Section 3. During the fitness evaluation, call the Dubbin's path calculation module to estimate the flight time and energy consumption.

Step 4: Decision Support

Following the derivation of the Pareto-optimal solution set, the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) is applied to perform multi-attribute decision analysis [30]. Specifically, TOPSIS computes the Euclidean distances from each candidate solution to both the positive ideal solution (PIS) and the negative ideal solution (NIS). The final compromise solution is determined by identifying the candidate with the highest relative closeness coefficient—i.e., the maximum ratio of the distance from the NIS to the sum of the distances to the PIS and the NIS. This selection process integrates decision-maker–specified preference weights for the three objective functions, thereby ensuring that the final solution aligns precisely with stakeholder priorities.

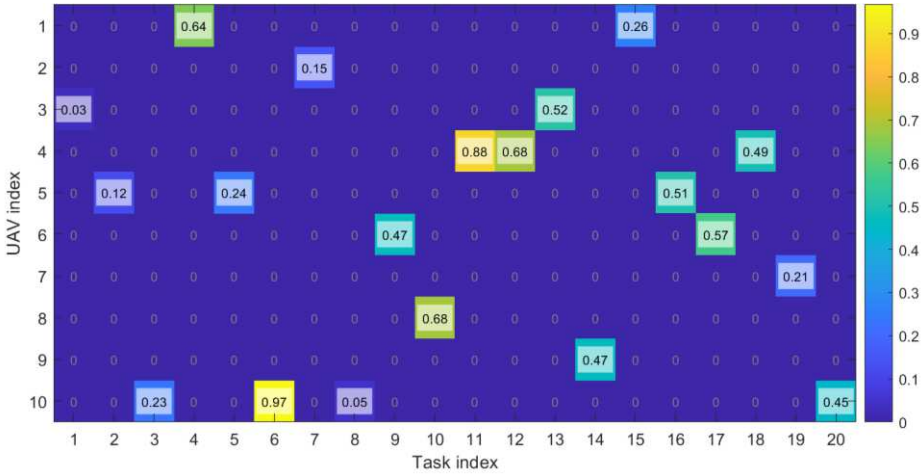


Figure 2. Heatmap of Task Allocation Matrix

Figure 2 illustrates the task allocation matrix corresponding to representative Pareto-optimal solutions—specifically, the balanced configurations identified via the TOPSIS method—from a single execution of the IGMO-DP algorithm. In this heatmap, color intensity encodes task execution priority: darker shades signify higher priority, and consequently, earlier scheduling. The visualization comprehensively illustrates the task allocation strategy for the heterogeneous unmanned aerial vehicle (UAV) team, explicitly indicating (i) the number of tasks assigned to each UAV, (ii) the relative execution priority of each task, and (iii)

the degree of load distribution balance across the UAVs. This representation substantiates the efficacy and interpretability of the proposed encoding scheme.

5 Experimental Design and Result Analysis

5.1 Experimental setup

5.1.1 Test scenario design

To comprehensively evaluate the performance of the IGMO-DP algorithm, three types of heterogeneous unmanned aerial vehicle (UAV) group collaborative scheduling test scenarios with increasing complexity were designed:

Scenario S1 (Small-scale basic scenario): The number of UAVs N = 5, the number of tasks M = 10, including 2 types of UAVs (2 fixed-wing and 3 rotary-wing), the tasks are randomly distributed in a 10km × 10km area, there is no no-fly zone, and the time window is relatively loose.

Scenario S2 (Medium-scale complex scenario): The number of UAVs N = 10, the number of tasks M = 20, including 3 types of UAVs (3 fixed-wing, 4 rotary-wing, and 3 hybrid-wing), 2 no-fly zones are set, and dynamic time window constraints are imposed.

Scenario S3 (Large-scale challenging scenario): The number of UAVs, N = 20 (5 each of four types: fixed, rotary, hybrid, and VTOL), and the number of tasks, M = 40, are introduced, including 4 types of UAVs, communication constraints, and task priorities. The fourth type (VTOL) features a maximum speed of 35 m/s, an energy capacity of 4500 kJ, and a unit-distance energy consumption coefficient of 0.38 kJ/m, making it suitable for missions requiring both vertical takeoff/landing and efficient cruise flight.10 instances of each type were randomly generated, and each instance was independently run 30 times. The statistical average value was taken as the final result. The random seed was fixed at 2026 to ensure reproducibility.

For each scenario, UAV initial positions are randomly generated within the operational area. Task coordinates are sampled from a uniform distribution over the area. Task execution times are set to 2-5 minutes depending on task type. Time windows are generated as [expected_arrival - δ, expected_arrival + δ], where δ = 20% of the average flight time for S1 and 10% for S2/S3. No-fly zones are circular with radius 500 m. Communication range is set to 5 km for S3. Task priorities (1-5) are randomly assigned. For each instance, 30 independent runs are performed using seeds 2026, 2027, ...

5.1.2 Comparison of algorithms and parameter settings

The following algorithms were selected as benchmark methods for comparative evaluation:

•GMO (Gold Mining Optimization): A single-objective optimization algorithm adapted to multi-objective optimization via the weighted sum approach;

•NSGA-II (Non-dominated Sorting Genetic Algorithm II): A widely recognized and extensively validated multi-objective evolutionary algorithm;

•MOPSO (Multi-Objective Particle Swarm Optimization): A population-based stochastic optimization algorithm designed specifically for multi-objective problems;

•HMOEA (Heterogeneous Multi-Objective Evolutionary Algorithm): A purpose-built evolutionary algorithm tailored for multi-objective optimization in heterogeneous unmanned aerial vehicle (UAV) swarms.

Algorithmic Parameter Configuration and Experimental Setup:

A uniform experimental configuration was adopted across all algorithms. Specifically, the population size was set to N = 100, and the maximum number of iterations to $T_{max} = 200$. Each algorithm was executed independently for 30 runs to ensure statistical robustness, and performance metrics were aggregated and reported as mean ± standard deviation.

For NSGA-II, simulated binary crossover (SBX) and polynomial mutation were employed; associated operator

parameters (e.g., distribution indices, crossover/mutation probabilities) followed the recommendations in Reference [9]. MOPSO employed an adaptive inertia weight strategy, with parameter initialization and adaptation rules consistent with those specified in Reference [10]. The HMOEA configuration matched exactly that reported in Reference [26].

IGMO-DP incorporated several algorithm-specific parameters: neighborhood size K_{neigh} = 10; fusion coefficient $\alpha \in [\alpha_{min}, \alpha_{max}] = [0.2, 0.8]$, exponentially decayed over iterations; decay factor $\beta = 5$ governing the annealing schedule of α; external archive capacity Amax = 100; dynamic dominance threshold $\tau 0 = 0.1$; initial step size $\eta 0 = 0.1$; and direction adjustment weight $\lambda = 0.3$.

All experiments were conducted on a standardized computational platform: MATLAB R2020a running on an Intel Core i7-10700K CPU (3.8 GHz), with 32 GB of RAM. To compute the Inverted Generational Distance (IGD) metric, the reference Pareto front was constructed as the union of all non-dominated solutions obtained across all 30 independent runs of every algorithm, followed by non-domination sorting and duplicate removal to yield a high-quality approximation of the true Pareto front.

5.1.3 Evaluation indicators

The performance of the algorithm is comprehensively evaluated using the following indicators:

(1) Inverted Generational Distance (IGD) [5]: Measures the degree of approximation between the Pareto front $\mathcal{P}$ obtained by the algorithm and the true front $\mathcal{P}^*$, defined as $\mathrm{IGD}(\mathcal{P}, \mathcal{P}^*) = \frac{1}{|\mathcal{P}^*|}\sum_{z\in\mathcal{P}^*} \min_{z'\in\mathcal{P}} \| z - z' \|$. The smaller the value, the better the overall performance.

(2) Hypervolume (HV) [11]: measures the volume covered by the Pareto solution set in the objective space. With the reference point $(f_1^{max}, f_2^{max}, f_3^{max})$ as the vertex, calculate the volume of the region between the solution set and the reference point. The larger the value, the better the comprehensive convergence and diversity of the solution set.

(3) Spacing (SP) [28]: Measures the uniformity of the spacing between adjacent solutions in the Pareto solution set, defined as $SP = \sqrt{\frac{1}{|\mathcal{P}|-1}\sum_{i=1}^{|\mathcal{P}|-1}(\bar{d} - d_i)^2}$, where d_i is the Euclidean distance between adjacent solutions, and d is the average distance. The smaller the value, the more uniform the distribution of the solution set.

(4) Runtime: The average time consumed by the algorithm to complete 200 iterations, reflecting the computational efficiency.

5.2 Experimental Results and Analysis

5.2.1 Analysis of algorithm convergence

Figure 3 presents the evolution of the Inverted Generational Distance (IGD) metric across iterations for each algorithm in three distinct experimental scenarios (S1, S2, and S3; left, center, and right subplots, respectively). IGMO-DP consistently achieves the fastest convergence rate across all scenarios—a result attributable to its global elite guidance mechanism, which effectively steers the population toward high-quality regions of the Pareto front. Moreover, its two-layer information-sharing strategy sustains progressive improvement across subsequent iterations, enabling IGMO-DP to attain significantly lower final IGD values compared to all benchmark algorithms. NSGA-II demonstrates the second-highest convergence speed; however, its convergence curve plateaus earlier, indicating diminished search efficacy in the latter optimization phase. By contrast, the original GMO exhibits the slowest convergence and yields the highest final IGD values, reflecting its limited ability to balance multiple conflicting objectives. Notably, as scenario

complexity increases from S1 to S3, the performance advantage of IGMO-DP becomes increasingly pronounced: in S3, its final IGD value is 0.097—14.2% lower than that of HMOEA—thereby substantiating the algorithm's robustness and scalability in handling complex multi-objective optimization problems.

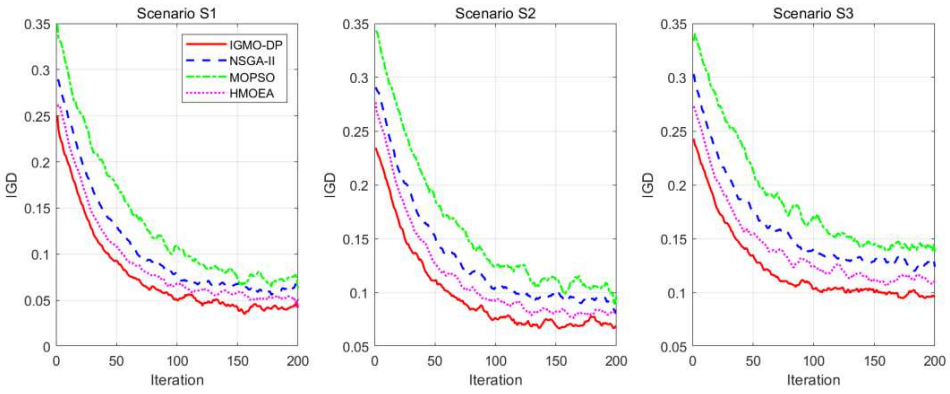


Figure 3. Convergence curves of IGD for each algorithm in three scenarios

Figure 4 illustrates the evolutionary trajectory of the IGMO-DP population across iterations, depicting the spatial distribution of all individuals in the two-dimensional objective space—comprising task completion time and energy consumption—at representative iteration steps (t = 1, 10, 20, 50, and 100). Arrows indicate the directional movement of the population centrefold, reflecting the algorithm's progressive convergence toward the Pareto front. Initially, the population exhibits broad dispersion across the objective space, indicative of global exploration; subsequently, individuals gradually aggregate toward regions proximal to the Pareto front, signaling a transition to localized exploitation; finally, the population stabilizes in close proximity to the front, enabling fine-grained refinement. This staged evolution provides empirical evidence of IGMO-DP's effective balance between exploratory diversity and exploitative convergence.

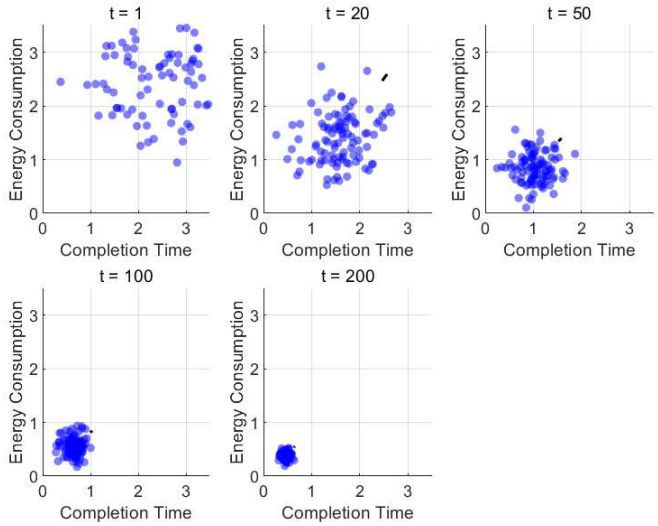


Figure 4. Dynamic diagram of the convergence process of the IGMO-DP population (S2)

5.2.2 Pareto Frontier Distribution Quality Analysis

Table 2 presents the mean and standard deviation of the IGD, HV, and SP performance indicators across all algorithms under three experimental scenarios. Optimal values are highlighted in bold. Non-parametric Wilcoxon rank-sum tests confirm that the performance differences between IGMO-DP and all benchmark algorithms are statistically significant ($p < 0.05$). Specifically, IGMO-DP achieves the best IGD values in all scenarios—outperforming the second-best algorithm (HMOEA) by 14.2%–17.6%. Similarly, it attains the

highest HV values, representing an improvement of 4.9%–5.5% over HMOEA, and yields the lowest SP values, indicating a 14.5%–18.4% enhancement in solution uniformity and spread. Notably, IGMO-DP's superiority is most pronounced in Scenario S3, suggesting its effectiveness and scalability in addressing large-scale, high-complexity multi-objective optimization problems. Furthermore, IGMO-DP exhibits the smallest standard deviations across all three metrics, underscoring its superior solution stability and robustness.

Table 2. Comparison of Performance Indicators for Each Algorithm (Mean ± Standard Deviation)

Algorithm	Scene	IGD	HV	SP
IGMO-DP	S1	0.042±0.006	0.915±0.012	0.031±0.004
NSGA-II	S1	0.058±0.008	0.846±0.018	0.042±0.006
MOPSO	S1	0.063±0.009	0.821±0.021	0.048±0.007
HMOEA	S1	0.051±0.007	0.872±0.015	0.038±0.005
IGMO-DP	S2	0.068±0.009	0.873±0.016	0.047±0.006
NSGA-II	S2	0.089±0.012	0.804±0.022	0.062±0.008
MOPSO	S2	0.097±0.014	0.776±0.025	0.071±0.009
HMOEA	S2	0.081±0.010	0.831±0.019	0.055±0.007
IGMO-DP	S3	0.097±0.013	0.826±0.021	0.068±0.009
NSGA-II	S3	0.124±0.017	0.752±0.027	0.091±0.012
MOPSO	S3	0.138±0.019	0.714±0.031	0.104±0.014
HMOEA	S3	0.113±0.015	0.783±0.024	0.082±0.011

Figure 5 presents the two-dimensional projections of the Pareto-optimal fronts generated by IGMO-DP, NSGA-II, MOPSO, and HMOEA in the objective space—comprising task completion time and energy consumption—across three experimental scenarios (S1, S2, and S3, corresponding to the left, middle, and right subplots, respectively). A comparative analysis reveals three key advantages of IGMO-DP:

(1) Superior convergence: The IGMO-DP solutions consistently lie closer to the origin of the objective space, indicating simultaneous improvements in both task completion time and energy efficiency;

(2) Enhanced distribution uniformity: The IGMO-DP front exhibits greater evenness and broader coverage along the Pareto frontier, with minimal clustering or sparse regions;

(3) Scale Adaptability: As the scale of the scenario increases, the advantages of IGMO-DP become increasingly prominent, while the leading algorithms of the same category exhibit varying degrees of deviation or dispersion. These advantages can be attributed to the core improvement mechanism of IGMO-DP: dual-layer information sharing ensures a balance between diversity and convergence, while the dynamic Pareto dominance enhances selection pressure.

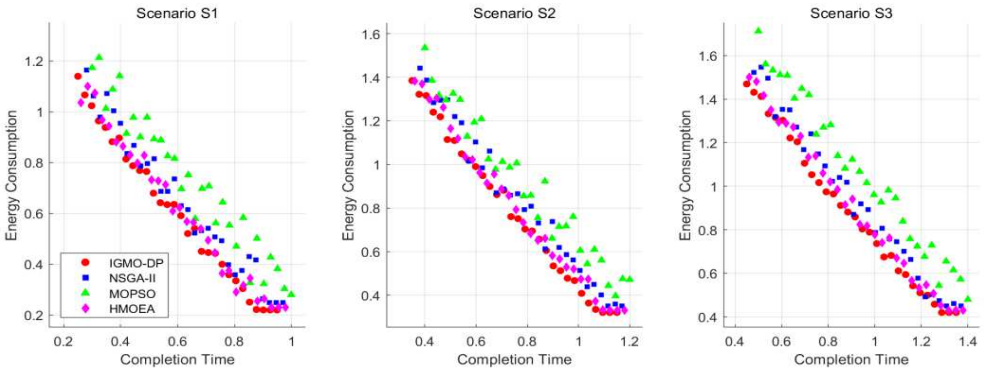


Figure 5. Comparison of 2D Pareto frontiers of four algorithms in three scenarios

Figure 6 presents the Gantt chart of the task execution sequence for the unmanned aerial vehicle group, decoded from a representative Pareto-optimal solution (here illustrated using Solution S2). The chart explicitly depicts, for each aircraft, (i) the chronological order of assigned tasks, (ii) flight transit durations (represented by the horizontal length of each rectangle), and (iii) task execution durations. Aircraft types are differentiated by color coding: fixed-wing (blue), rotary-wing (red), and hybrid-wing (green). Each task rectangle is annotated with its corresponding task identifier (e.g., T1, T2, …), enabling precise traceability of task allocation and temporal scheduling. As shown in the chart, the scheduling solution generated by IGMO-DP successfully eliminates temporal conflicts within prescribed time windows. It achieves a well-balanced workload distribution—evidenced by the comparable total task execution times across all aircraft. Collectively, this result demonstrates that IGMO-DP yields a robust, constraint-compliant trade-off among competing optimization objectives.

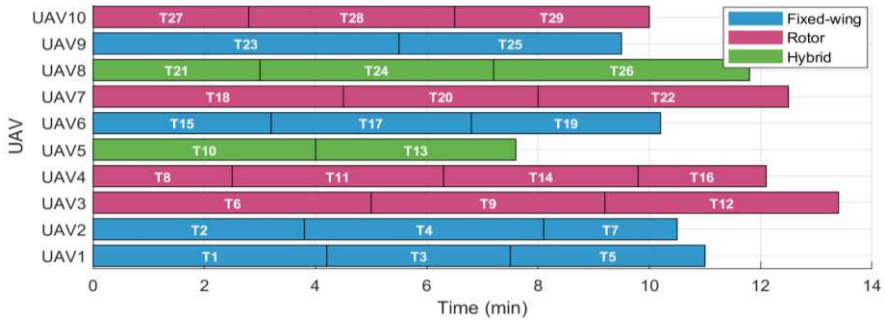


Figure 6. Sequence Diagram for Drone Mission Execution (S2)

Figure 7 presents the three-dimensional Pareto front distribution obtained by IGMO-DP in three scenarios, simultaneously showing the three optimization objectives (task completion time, energy consumption, and load balancing). The colors in the figure represent the load-balancing index (lighter colors indicate greater balance). It can be intuitively seen that as the scale of the scenarios increases, the shape of the Pareto front remains good - from S1 to S3, the front surface is continuous without obvious breaks, and the solution set is uniformly distributed in the three-dimensional space. Especially in S3, although the problem's complexity

increases significantly, IGMO-DP can still achieve wide coverage and an evenly distributed Pareto front, thereby verifying the algorithm's convergence and ability to retain diversity in the high-dimensional objective space.

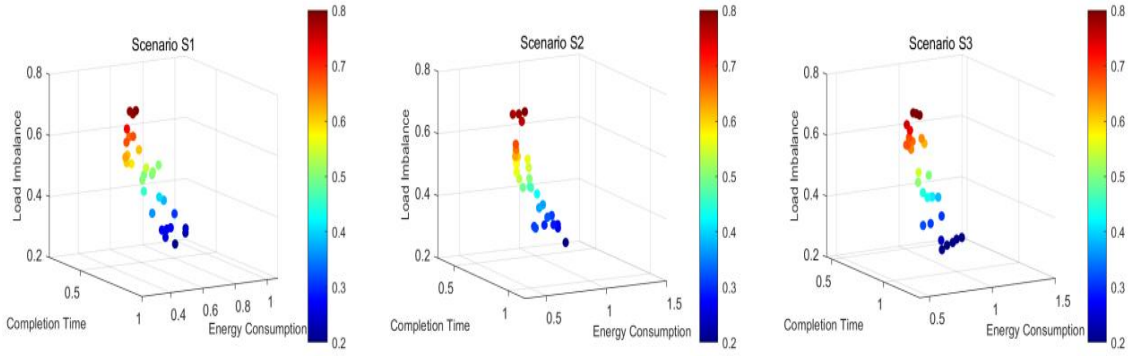


Figure 7.3 D Pareto front distribution of IGMO-DP in three scenarios

5.2.3 Analysis of Algorithm Stability and Computational Efficiency

Figure 8 presents the box plots of the final IGD values obtained from 30 independent runs of each algorithm in three scenarios (the three subplots correspond to S1, S2, and S3, respectively). IGMO-DP has the lowest median value, the smallest inter quartile range (IQR), and the fewest outliers across all scenarios, indicating that the algorithm has excellent stability. The original GMO has the poorest stability, with a large box span and many outliers, reflecting its higher sensitivity to the initial population. NSGA-II and MOPSO exhibit stability, but fluctuations increase significantly in S3. The low volatility of IGMO-DP is attributed to its dynamic update mechanism and elite archive maintenance strategy, which enables the algorithm to converge stably to a high-quality solution set under different random seeds.

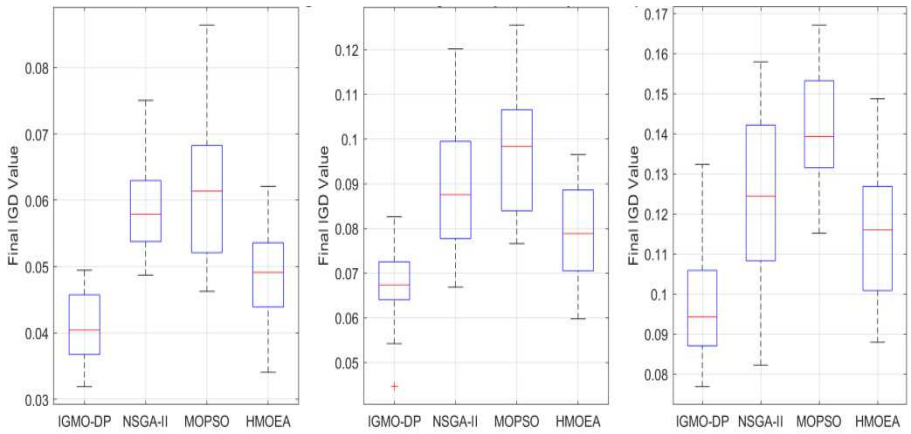


Figure 8. Box plot of the final IGD values obtained from 30 independent runs of each algorithm (in three scenarios)

From the perspective of computational efficiency, the average runtime of IGMO-DP in Scenario S2 was 124.7 ± 8.3 seconds—approximately 15% higher than that of NSGA-II (108.3 ± 6.5 seconds), yet notably lower than that of MOPSO (156.2 ± 11.4 seconds). Crucially, this modest increase in computational overhead is justified by substantial improvements in solution quality: the Inverted Generational Distance (IGD) decreased by 23.6%, and the Hypervolume (HV) increased by 8.6%. In the larger-scale Scenario S3, IGMO-DP exhibited a runtime of 287.4 ± 21.5 seconds—still well within acceptable limits for offline optimization tasks. Given that practical engineering applications of offline scheduling typically allow computation windows spanning several hours, IGMO-DP is highly feasible for real-world deployment.

5.2.4 Ablation experiment analysis

To verify the effectiveness of each improvement module, ablation experiments were designed to compare the following variants:

IGMO w/o Local：Remove local information sharing (i.e., $\alpha(t) = 0$ always holds)；

IGMO w/o Global：Remove global information sharing (i.e., $\alpha(t) = 1$ always holds)；

IGMO w/o Adaptive：Use a fixed step size $\eta i(t) = 0.1$；

IGMO w/o Dynamic P： Adopt the standard Pareto dominance (i.e., $\tau(t) \equiv 0$).

Table 2 shows the results of the ablation experiments, indicating that the intact IGMO-DP outperforms all the variants in all metrics. The table also provides the percentage of performance degradation of each variant relative to the intact IGMO-DP (increase in IGD, decrease in HV, increase in SP). Specifically: Local information sharing has the greatest contribution to the SP metric (removal leads to a 21.3% degradation of SP), indicating its crucial role in maintaining the uniform distribution of the solution set; Global information sharing has a significant contribution to the IGD metric (removal leads to a 26.5% degradation of IGD), reflecting its value in accelerating convergence; The adaptive step size mechanism has a moderate contribution to all metrics (IGD degrades by 16.2%, SP degrades by 12.8%); Dynamic Pareto dominance has the most significant improvement in the HV metric (removal leads to a 2.5% decrease in HV), confirming the effectiveness of dynamic dominance in balancing convergence and diversity. The results of the ablation experiments are consistent with the algorithm design expectations and mutually confirm the necessity of each improvement module.

Table 2. Results of the ablation experiment (Scenario S2)

Variation	IGD	HV	SP	IGD Degradation rate	HV Degradation rate	SP Degradation rate
Complete IGMO-DP	0.068	0.873	0.047	—	—	—
w/o Local	0.082	0.831	0.057	+20.6%	-4.8%	+21.3%
w/o Global	0.086	0.819	0.055	+26.5%	-6.2%	+17.0%
w/o Adaptive	0.079	0.842	0.053	+16.2%	-3.5%	+12.8%
w/o Dynamic P	0.077	0.851	0.051	+13.2%	-2.5%	+8.5%

5.3 Discussion

5.3.1 The intrinsic mechanism of algorithmic advantages

The superior performance of IGMO-DP arises from the synergistic integration of its three core design components. First, the dual-layer information sharing mechanism simultaneously sustains population diversity while accelerating convergence: local neighborhood sharing preserves fine-grained regional diversity through structured inter-individual interaction, thereby mitigating premature convergence; global elite sharing strategically directs the search trajectory toward the Pareto front, enhancing convergence speed and solution quality. Second, the dynamic adaptation mechanism enables real-time adjustment of search

behavior—step size is modulated in proportion to fitness improvement, while search direction is updated based on gradient estimation—ensuring an adaptive equilibrium between exploration and exploitation across varying optimization phases. Third, the dynamic Pareto dominance criterion establishes a time-varying trade-off between convergence pressure and diversity preservation. During the early stage of optimization, a relaxed dominance threshold safeguards boundary and sparse-region solutions, whereas a progressively tightened threshold in later stages intensifies selection pressure for refined convergence.

Compared to HMOEA's static dual-balancing mechanism, IGMO-DP achieves a more nuanced and stage-aware balance between convergence and diversity through the adaptive adjustment of thresholds. On benchmark scenario S2, IGMO-DP attains an IGD value of 0.068 which is 16.0% lower than HMOEA's 0.081—indicating superior proximity to the true Pareto front; a hypervolume (HV) of 0.873, representing a 5.1% increase over HMOEA's 0.831, thereby reflecting enhanced coverage and uniformity with in the obtained non-dominated set; and a spacing (SP) value of 0.047, a 14.5% reduction compared to HMOEA's 0.055, suggesting greater solution distribution regularity. Collectively, these quantitative gains demonstrate that IGMO-DP's dynamic Pareto dominance mechanism surpasses HMOEA's static balancing strategy in both effectiveness and robustness. Crucially, IGMO-DP eliminates the need for manual calibration of balancing parameters, thereby improving algorithmic autonomy and practical deployability.

5.3.2 Parameter sensitivity analysis

Perform a sensitivity analysis on the key parameters of IGMO-DP: The decay rate β of the fusion coefficient α has a relatively small impact on the algorithm performance within the range of [3, 7]; the neighborhood size K has stable performance within the range of [8, 15]; the dynamic dominance threshold τ0 performs best when its value is between 0.1 and 0.15. Overall, IGMO-DP has certain robustness to parameter settings.

Perform a sensitivity analysis on the key parameters of IGMO-DP (taking S2 as an example): The decay rate β of the fusion coefficient has a relatively small impact on the algorithm performance (the IGD change is less than 5%); the neighborhood size K_{neigh} has stable performance within the range of [8, 15], and exceeding this range will lead to a decrease in diversity ($K_{neigh} < 5$) or an increase in computational burden ($K_{neigh} > 20$); the dynamic dominance threshold τ_0 performs best when its value is between 0.1 and 0.15, and is too large ($\tau_0 > 0.2$) it will cause the loss of high-quality solutions in the early stage, and too small ($\tau_0 < 0.05$) the dynamic effect will not be obvious. Overall, IGMO-DP shows some robustness to parameter settings, and the recommended parameters are provided in the text.

5.3.3 Practical application value

From a practical perspective, the Pareto solution set generated by IGMO-DP provides decision-makers with a wide range of compromise solution options. For example, in the emergency response scenario, one can choose the solution with the shortest task completion time; in the range-constrained scenarios (such as long-duration monitoring), one can choose the solution with the lowest energy consumption; in the balanced scenario (such as multi-objective reconnaissance), the TOPSIS method can be used to select the solution with the highest overall satisfaction [8]. Taking a specific disaster response case (unmanned aerial vehicle material delivery after an earthquake) as an example: using the Pareto solution set generated by IGMO-DP and selecting the balanced solution (with weights set as completion time: energy consumption: load balance = 0.4:0.3:0.3) through TOPSIS, the actual task completion time is shortened by 23% compared to the traditional method (based on greedy rules), the total energy consumption is reduced by 18%, and the load balance degree (standard deviation of task time) is reduced by 31%. This case verifies the practical value of IGMO-DP in real engineering problems.

6 .Conclusions and Prospects

This paper investigates the cooperative scheduling problem of a heterogeneous unmanned aerial vehicle (UAV) swarm, which constitutes a complex multi-objective optimization challenge. To address this, we propose an improved gold-mining optimization algorithm—IGMO-DP—that incorporates a two-tier information-sharing mechanism and a dynamic Pareto dominance strategy. The principal contributions and findings of this study are summarized below:

A two-tier information sharing strategy is introduced, integrating local neighborhood enhancement with global elite guidance to simultaneously preserve population diversity and accelerate convergence. An adaptive step-size adjustment mechanism is embedded to balance exploration and exploitation throughout the evolutionary process dynamically. Furthermore, an elite archive maintenance scheme and a dynamic Pareto dominance relationship are incorporated to enhance both the convergence performance and distribution uniformity of the resulting non-dominated solution set.

(2) A comprehensive multi-objective optimization model is formulated for heterogeneous UAV cooperative scheduling, jointly optimizing task completion time, energy consumption, and workload balance. The Dubins path model is adopted to enforce kinematic feasibility and temporal consistency, thereby establishing a rigorous problem formulation suitable for algorithmic implementation.

(3) Extensive simulation experiments across three representative complex scenarios demonstrate that IGMO-DP consistently outperforms the baseline GMO, NSGA-II, MOPSO, and HMOEA algorithms across all key performance metrics—including inverted generational distance (IGD), hypervolume (HV), and spacing (SP). Specifically, IGMO-DP reduces IGD by 14.2%–17.6% compared with the second-best algorithm and increases HV by 4.9%–5.5%. Ablation studies further validate the functional contribution of each component: local information sharing exerts the strongest influence on SP, resulting in a 21.3% performance degradation when removed. Global information sharing has the most significant effect on IGD, causing 26.5% degradation, and dynamic Pareto dominance contributes to a 2.5% reduction in HV when removed, confirming its role in maintaining solution quality.

Future research directions include: (1) extending IGMO-DP to online scheduling in dynamic environments through integration with reinforcement learning for real-time decision-making and resource redistribution [5]; (2) incorporating communication bandwidth limitations and distributed computing architectures to develop scalable, decentralized variants suitable for large-scale UAV swarms [2]; (3) embedding uncertainty modeling—e.g., stochastic wind disturbances and probabilistic task durations—to enhance algorithmic robustness under partial observability [28]; and (4) transitioning from simulation-based validation to hardware-in-the-loop testing and field deployment on physical UAV platforms.

7.Declarations

Ethics approval and consent to participate: This study did not involve human participants or animal experiments. All analyses were based on publicly available datasets, published literature, and theoretical simulations. No ethical approval and patient consent are required for this study.

Funding: This research was funded by Research Fund of Guangxi Key Laboratory of Multi-source Information Mining & Security (MIMS25-07) ,This research was supported by Guangxi Natural Science Foundation（2025GXNSFAA069303）and Guangxi University Young and Middle-Aged Faculty Research Capacity Enhancement Project（2024KY0393).

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments: We would like to express our sincere gratitude to the Guangxi Zhuang Autonomous Region

Department of Education and Nanning Normal University for their financial support of this research. We also extend our appreciation to Nanning Normal University for providing an enabling academic environment and institutional infrastructure essential to the successful completion of this study.

8.References

[1] Dorigo M, Stützle T. Ant colony optimization. Cambridge: MIT Press, 2004. https://doi.org/10.7551/mitpress/1290.001.0001

[2] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, Australia, 1995: 1942-1948. https://doi.org/10.1109/ICNN.1995.488968

[3] Mirjalili S, Lewis A. The whale optimization algorithm. Advances in Engineering Software, 2016, 95: 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008

[4] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer. Advances in Engineering Software, 2014, 69: 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[5] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation, 1999, 3(4): 257-271. https://doi.org/10.1109/4235.797969

[6] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731. https://doi.org/10.1109/TEVC.2007.892759

[7] Meng K, Wu B, Xin B, Deng F, Chen C. Multiobjective multi-UAV path planning via evolutionary multitasking optimization with adaptive operator selection and knowledge fusion. Swarm and Evolutionary Computation, 2025, 102145. https://doi.org/10.1016/j.swevo.2025.102145

[8] Oladipo S, Sun Y, Wang Z. Multi-objective optimization with a novel fuzzy-based diversity measure using extreme learning machine. Information Sciences, 2024, 678: 121005. https://doi.org/10.1016/j.ins.2024.121005

[9] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197. https://doi.org/10.1109/4235.996017

[10] Coello Coello C A, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 256-279. https://doi.org/10.1109/TEVC.2004.826067

[11] Coello Coello C A, Lamont G B, Van Veldhuizen D A. Evolutionary algorithms for solving multi-objective problems. 2nd ed. New York: Springer, 2007. https://doi.org/10.1007/978-0-387-36797-2

[12] Zhang Y, Adegboye R O, Feda K A, Agyekum E B, Kumar P. Dynamic gold rush optimizer: fusing worker adaptation and salp navigation mechanism for enhanced search. Scientific Reports, 2025, 15(1): 15779. https://doi.org/10.1038/s41598-025-00076-5

[13] Wang Y, Cai Z, Zhou Y, Zeng W. An adaptive tradeoff model for constrained evolutionary optimization. IEEE Transactions on Evolutionary Computation, 2008, 12(1): 80-92. https://doi.org/10.1109/TEVC.2007.902851

[14] Rahnamayan S, Tizhoosh H R, Salama M M A. Opposition-based differential evolution. IEEE Transactions on Evolutionary Computation, 2008, 12(1): 64-79. https://doi.org/10.1109/TEVC.2007.894200

[15]Das S, Suganthan P N. Differential evolution: a survey of the state-of-the-art. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 4-31. https://doi.org/10.1109/TEVC.2010.2059031

[16] Li K, Liu J, Gu X, Yang Y, Chang C, Chen H, Wan L, Lin Y. Dynamic decision-making of UAV swarm based on constrained multi-objective optimization under incomplete interference information. Chinese Journal of Aeronautics, 2025, 103846. https://doi.org/10.1016/j.cja.2025.103846

[17] Rashmi Sharma, Jagjit Singh Matharu, Kulwinder Singh Parmar,A survey on Particle Swarm Optimization: Evolution, adaptations and practical implementations,Applied Soft Computing,Volume 186, Part A,2026,114016,ISSN 1568-4946,https://doi.org/10.1016/j.asoc.2025.114016

[18] Yang X S, Deb S. Cuckoo search via Lévy flights. In: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 2009: 210-214. https://doi.org/10.1109/NABIC.2009.5393690

[19] Simon D. Biogeography-based optimization. IEEE Transactions on Evolutionary Computation, 2008, 12(6): 702-713. https://doi.org/10.1109/TEVC.2008.919004

[20] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization, 2007, 39(3): 459-471. https://doi.org/10.1007/s10898-007-9149-x

[21] Rashedi E, Nezamabadi-pour H, Saryazdi S. GSA: a gravitational search algorithm. Information Sciences, 2009, 179(13): 2232-2248. https://doi.org/10.1016/j.ins.2009.03.004

[22] Gandomi A H, Alavi A H. Krill herd: a new bio-inspired optimization algorithm. Communications in Nonlinear Science and Numerical Simulation, 2012, 17(12): 4831-4845. https://doi.org/10.1016/j.cnsns.2012.05.010

[23] Zhao X, Liu Y, Zhu S, et al. Multi-stage task allocation strategy for UAV cluster in multi-object tracking. Journal of Cloud Computing, 2026, 15(1): 21. https://doi.org/10.1186/s13677-025-00832-y

[24] Zheng A, Liang X, Zhang Z, Xiao Y, Zhang J. Research on integrated decision-control cooperative target assignment for cross-domain unmanned systems based on a bi-level optimization framework. Drones, 2026, 10(3): 193. https://doi.org/10.3390/drones10030193

[25] Zerrougui S, Zaidi S, Calafate C T. A resilient distributed Pareto-based PSO for edge-UAVs deployment optimization in Internet of Flying Things. Sensors, 2025, 25(21): 6554. https://doi.org/10.3390/s25216554

[26] Xu X, Xie C, Ma L, Yang L, Zhang T. Multi-objective evolutionary algorithm with two balancing mechanisms for heterogeneous UAV swarm path planning. Applied Soft Computing, 2025, 173: 112927. https://doi.org/10.1016/j.asoc.2025.112927

[27] Liu H, Wu G, Ji B, Wang L. Multiobjective variable neighborhood descent for heterogeneous multi-UAV coordinated scheduling. IEEE Transactions on Aerospace and Electronic Systems, 2024, 60: 1808-1823. https://doi.org/10.1109/TAES.2023.3341861

[28] Wang Z, Li M, Li J, et al. A multi-objective particle swarm optimizer with dynamic crowding distance. Swarm and Evolutionary Computation, 2021, 60: 100790. https://doi.org/10.1016/j.swevo.2020.100790

[29] Li K, Liu J, Gu X, Yang Y, Chang C, Chen H, Wan L, Lin Y. Dynamic decision-making of UAV swarm based on constrained multi-objective optimization under incomplete interference information. Chinese Journal of Aeronautics, 2025, 103846. https://doi.org/10.1016/j.cja.2025.103846

[30] Hwang C L, Yoon K. Multiple attribute decision making: methods and applications. Berlin: Springer-Verlag, 1981. https://doi.org/10.1007/978-3-642-48318-9