



A risk-aware improved grey wolf optimizer algorithm for multi-UAV task allocation in spatially constrained environments

Xueru Miao, Zhaoxian Zhang, Yuqi Wang, Bo Li

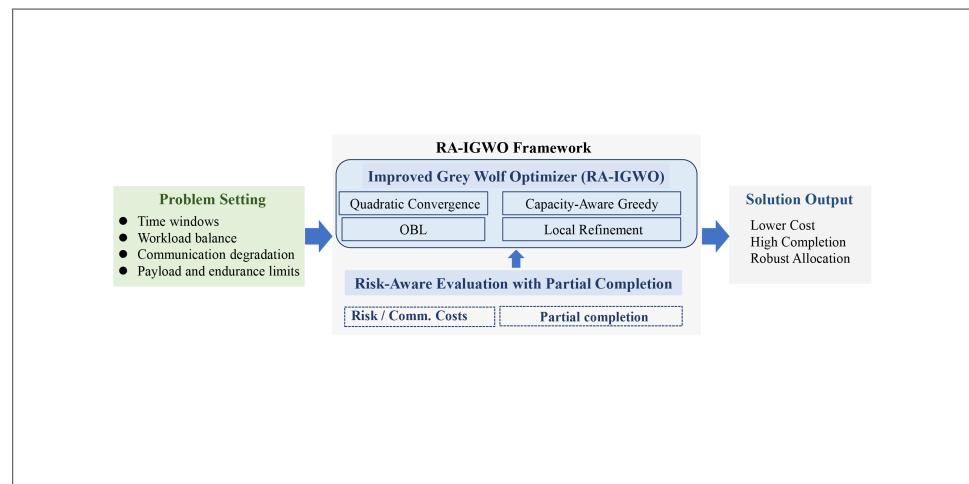
Keywords:

Multi-UAV systems, cooperative task allocation, risk-aware improved grey wolf optimizer, opposition-based learning

Citation: Miao, X.; Zhang, Z.; Wang, Y.; Li, B. A risk-aware improved grey wolf optimizer algorithm for multi-UAV task allocation in spatially constrained environments. *Complex Eng. Syst.* 2026, 6, 17. <https://dx.doi.org/10.20517/ces.2026.22>

Received: 26 Jul 2026
First Decision: 20 Aug 2026
Revised: 10 Sep 2026
Accepted: 17 Sep 2026
Published: 29 Oct 2026

Academic Editor:
 Wenwu Yu
Copy Editor:
 Fangling Lan
Production Editor:
 Fangling Lan



Abstract

This paper investigates risk-aware task allocation for heterogeneous multi-UAV systems under coupled communication, time-window, workload, spatial-risk, payload, and endurance constraints. Existing formulations may exclude useful traversable routes or underestimate executable service by treating hazardous regions as hard barriers or using all-or-nothing feasibility evaluation. To address these limitations, a risk-aware improved Grey Wolf Optimizer (RA-IGWO) is developed within a partial-completion framework. Risk-prone and communication-degraded regions are modeled as traversable soft-cost areas, while physical obstacles are explicitly penalized and additionally examined under hard-obstacle enforcement. RA-IGWO integrates capacity-aware initialization, opposition-based learning, decoded-space local refinement, and accept-only-improved population updates. The quadratic schedule is retained only as an exploration policy rather than an independently beneficial component. All algorithms are evaluated under the same global number of fitness evaluations budget, with every candidate evaluation counted toward that budget. Experiments on independently generated instances show that RA-IGWO achieves lower objective values than five population-based baselines and remains competitive with a recent discrete

Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China.

Correspondence to: Prof. Bo Li, Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China. E-mail: libo@shmtu.edu.cn

adaptive large neighborhood search benchmark method. Robustness analyses show that the conclusions are not tied to a single decoder, obstacle treatment, or objective setting.

1. INTRODUCTION

Unmanned aerial vehicle (UAV) swarms are increasingly used in large-scale reconnaissance, medical-supply delivery, environmental monitoring, and post-disaster response^[1,2]. Compared with homogeneous fleets, heterogeneous UAV fleets combine different cruise speeds, payload capacities, and endurance limits, thereby supporting broader coverage and diverse payload requirements. These advantages depend on a central decision problem: allocating spatially distributed tasks to available UAVs while satisfying physical and operational constraints. Multi-UAV task allocation (MUTA) is NP-hard, and its solution complexity grows rapidly with the numbers of tasks and UAVs^[3,4].

Existing MUTA methods can be broadly grouped into three paradigms. Centralized exact methods, including mixed-integer programming formulations, can provide optimal solutions for small instances, but their computational burden increases rapidly with problem size and limits their use in time-sensitive missions. The consensus-based bundle algorithm is representative of decentralized market and consensus methods. During bundle construction, agents select tasks independently according to local marginal gains; during consensus, conflicts are resolved through asynchronous exchange of winner lists and timestamps without a central computing node^[5,6]. Recent extensions have considered heterogeneous swarms with task-coupling constraints^[7,8] and dynamic reallocation under emergent task changes^[8]. Although these methods are scalable and robust, their solution quality depends strongly on bidding and scoring rules, and degraded communication can reduce coordination quality.

Swarm intelligence and evolutionary metaheuristics provide another approach to task allocation, including genetic algorithms^[9,10], particle swarm optimization (PSO)^[11], differential evolution^[12], ant colony optimization (ACO)^[13], and artificial bee colony (ABC) optimization^[14]. These methods do not require gradient information and can accommodate mixed discrete and continuous structures through suitable encoding and decoding. Recent examples include multi-objective ACO for heterogeneous UAV task allocation^[15] and reinforcement-learning-enhanced ABC for cooperative unmanned-helicopter scheduling^[16]. The no-free-lunch theorem implies that no metaheuristic dominates across all problem classes^[17], motivating algorithms that exploit the structure of a particular task domain. Recent studies have increasingly considered the coupling of UAV decision-making with communication, energy, and low-altitude obstacle constraints. For example, secure aerial communication in low-altitude intelligent networks with mixed obstacles has been investigated in^[18], while joint task allocation and power optimization has been formulated as an overlapping coalition-formation problem in^[19]. Furthermore, adaptive operator selection using a dueling deep Q-network has been studied in evolutionary optimization^[20]. These studies advance communication optimization, decision frameworks, or adaptive operator selection, but do not explicitly address the coupled effects of risk exposure, capacity-limited execution, and many-to-one decoded allocation structures in risk-aware MUTA.

The Grey Wolf Optimizer (GWO)^[21] has received substantial attention because of its simple control structure and multi-leader search mechanism. Its compact formulation facilitates hybridization^[22] and has supported applications in multi-robot search and exploration^[23]. However, standard GWO may converge prematurely in rugged and strongly constrained search spaces. Therefore, several GWO variants have been developed to improve convergence and population diversity in complex engineering optimization problems^[24,25,26]. In MUTA, this challenge is intensified by discrete decoding boundaries and the interaction among capacity-related service loss, route exposure, and task-omission penalties. Existing variants introduce mechanisms such as Lévy flight perturbation^[27], chaotic initialization^[25,28], opposition-based learning (OBL)^[29,30], dimension-learning-based hunting^[24], PSO-GWO hybridization^[31,32], and GWO-symbiotic organisms search hybridization for UAV path

planning^[33]. Most of these improvements target general continuous optimization and primarily modify search operators or population diversity. Risk-aware MUTA additionally requires mechanisms that account for discrete decoding, capacity-limited service, and allocation-level interactions.

Many existing MUTA studies adopt an all-or-nothing feasibility rule, under which a route is considered infeasible once its payload or endurance limit is exceeded. This treatment can over-penalize marginal violations because tasks completed before the first violation may remain physically executable. The present study instead evaluates the feasible prefix of each decoded route. A second methodological issue concerns computational fairness. Metaheuristics with embedded local search or auxiliary candidate generation may consume more fitness evaluations than conventional population updates. Comparisons based only on an equal number of generations can therefore provide such methods with a larger effective evaluation budget. To avoid this bias, all algorithms terminate under the same global number of fitness evaluation (NFE) budget.

The above analysis reveals three coupled difficulties in risk-aware MUTA. First, continuous metaheuristic updates can become ineffective after many-to-one integer decoding. Second, payload- and endurance-limited execution makes solution quality depend not only on task ownership but also on how partial service is evaluated. Third, risk and communication exposure must be balanced against task completion. These characteristics motivate an integrated design in which global exploration, structured initialization, decoded-space refinement, and risk-aware partial-completion evaluation address different but interacting aspects of the same allocation problem.

Based on the above analysis, this paper develops risk-aware improved Grey Wolf Optimizer (RA-IGWO) for risk-aware MUTA in complex environments. The framework is designed around the structure of the decoded allocation landscape rather than as a general-purpose collection of search operators. The main contributions are summarized as follows:

1. A risk-aware execution model is formulated in which risk-prone and weak-communication zones are treated as traversable soft-cost regions. Physical obstacles are explicitly distinguished and validated under both finite-penalty and hard-constraint settings. A partial-completion rule evaluates the executable prefix of each decoded UAV route under payload and endurance limits.
2. RA-IGWO is designed for the many-to-one decoded assignment landscape by combining capacity-aware greedy seeding and OBL with allocation-level load-aware local refinement and accept-only-improved population updates. Local refinement acts directly on discrete task ownership and can cross plateaus that remain unchanged under small continuous perturbations. The quadratic schedule is treated as a tested exploration policy.
3. A budget-consistent benchmarking protocol is established in which every candidate evaluation generated by population updates, OBL, or local refinement consumes one evaluation from the same global NFE budget. Then the comparisons are examined using paired multi-instance statistics and robustness analyses.

2. SYSTEM MODEL AND OPTIMIZATION FORMULATION

2.1 Mission model and assignment representation

Consider a heterogeneous fleet $\mathcal{U} = \{1, \dots, M\}$ of M UAVs and a set $\mathcal{T} = \{1, \dots, N\}$ of spatially distributed tasks. The mission environment is a square of side length s , and a single base is located at $\mathbf{p}_b = (s/2, 0)$ on the lower boundary. Each task $j \in \mathcal{T}$ is described by its position $\mathbf{p}_j \in [0, s]^2$, reward $r_j > 0$, payload demand $q_j > 0$, and latest service time τ_j . For tasks without a deadline, $\tau_j = \infty$. Each UAV $i \in \mathcal{U}$ is characterized by cruise speed v_i ,

Table 1. Principal notation

Symbol	Meaning
$\mathcal{U}, M$	Set of UAVs and its cardinality
$\mathcal{T}, N$	Set of tasks and its cardinality
$\mathbf{p}_b, \mathbf{p}_j$	Position of the base and of task j
r_j, q_j, τ_j	Reward, payload demand, and latest service time of task j
v_i, Q_i, L_i	Speed, payload capacity, and endurance of UAV i
$\mathcal{Z}^r, \mathcal{Z}^c, \mathcal{Z}^o$	Risk-prone, weak-communication, and obstacle zones
$\mathbf{x}, x_j$	Assignment vector; UAV assigned to task j
$\mathbf{z}, [\ell, u]$	Continuous search vector and per-coordinate bounds
$\mathcal{T}_i, \pi_i, S_i$	Tasks assigned to UAV i ; visit order; served prefix
Δ	Pairwise Euclidean distance matrix
$\mathbf{R}, \mathbf{H}$	Risk-overlap matrix and obstacle-intersection matrix
d_i	Cumulative outbound service distance of UAV i
T_j	Arrival time at served task j
$F(\mathbf{x})$	Scalarized objective to be minimized
g, G	Fitness-evaluation counter and evaluation budget
P	Population size
$a(g)$	GWO convergence factor
$\mathbf{c}_z, \rho_z$	Center and radius of zone z used in Eq. (2)
ρ_{cap}	Payload-based capacity ceiling on completion rate in Eq. (12)

payload capacity Q_i , and an endurance-distance budget L_i . Data-driven regression and inference methods have been used to model UAV battery-discharge behavior^[34]. In the present allocation model, these dynamics are abstracted into the maximum allowable flight-distance budget L_i .

The environment contains three types of circular zones. A zone z is specified by center $\mathbf{c}_z$ and radius ρ_z . The sets $\mathcal{Z}^r$, $\mathcal{Z}^c$, and $\mathcal{Z}^o$ denote risk-prone zones, weak-communication zones, and obstacle zones, respectively. Risk-prone and weak-communication zones are modeled as traversable soft-cost regions because they represent increased mission exposure or degraded link reliability rather than absolute no-fly restrictions. Physical obstacles have a different interpretation. In the default formulation, obstacle intersections receive a high finite penalty and are therefore strongly discouraged but not mathematically prohibited. A separate hard-obstacle mode treats an intersecting served segment as infeasible.

The formulation is based on five assumptions. First, each flight route is a straight segment traversed at constant speed v_i , so the time to reach a task equals the cumulative outbound service distance divided by v_i . Second, each task can be executed by at most one UAV. Third, payload consumption and flight distance accumulate along the route, and service duration is negligible relative to flight time. Fourth, the default endurance model considers the cumulative outbound service distance from the base. The return-guaranteed mode instead requires $d_i + \Delta_{\text{prev},j} + \Delta_{jb} \leq L_i$ before accepting task j , thereby reserving sufficient range to return to base; Section 4.9 evaluates this mode. Finally, all risk, communication, and obstacle maps are static and known. Principal notation is summarized in Table 1.

The decision variable is an assignment vector $\mathbf{x} = (x_1, \dots, x_N)$, with $x_j \in \mathcal{U}$. For population-based real-valued optimization, each individual is represented by a continuous vector $\mathbf{z} \in [\ell, u]^N$, where $\ell = 1$ and $u = M$. Rounding and clipping are commonly used when continuous optimization operators are applied to integer-valued decision variables^[35]. The decoding rule is

$$\mathbf{x} = \text{dec}(\mathbf{z}), \quad x_j = \text{clip}(\text{round}(z_j), 1, M), \quad (1)$$

where $\text{clip}(\cdot, 1, M)$ restricts the rounded value to $[1, M]$. This mapping places all population-based optimizers in a common Euclidean search space, while the fitness function evaluates decoded task assignments. The mapping is many-to-one: multiple continuous vectors may decode to the same integer assignment, creating flat regions for continuous position updates. Although this representation is retained to provide a common search interface for the continuous baselines, RA-IGWO does not rely exclusively on continuous perturbations because its local-refinement operator acts directly in the integer assignment space. The greedy seed solution in Section 3.2 is encoded into the $[\ell, u]^N$ space and subjected to uniform random jitter with an amplitude of 0.3. Boundary clipping is then applied to keep every coordinate within $[\ell, u]$.

Motivated by threat-perception-based UAV path planning, candidate paths are evaluated using predefined exposure and feasibility costs^[36]. Several geometric quantities that depend only on the static environment can therefore be precomputed. The augmented node set is $\mathcal{V} = \mathcal{T} \cup \{b\}$, where $|\mathcal{V}| = N + 1$. The pairwise distance matrix $\Delta \in \mathbb{R}^{(N+1) \times (N+1)}$ stores Euclidean distances $\Delta_{uv} = \|\mathbf{p}_u - \mathbf{p}_v\|_2$. The risk-overlap matrix $\mathbf{R}$ and obstacle-intersection matrix $\mathbf{H}$ are constructed simultaneously.

To calculate the elementwise values of $\mathbf{R}$ and $\mathbf{H}$, consider the flight segment connecting nodes u and v . Let $\mathbf{d} = \mathbf{p}_v - \mathbf{p}_u$ and $\mathbf{f} = \mathbf{p}_u - \mathbf{c}_z$. If $\|\mathbf{p}_v - \mathbf{p}_u\|_2 = 0$, the overlap is defined as zero. Otherwise, substituting $\mathbf{p}(\lambda) = \mathbf{p}_u + \lambda \mathbf{d}$ into the circle equation gives $A_q \lambda^2 + B_q \lambda + C_q = 0$, where $A_q = \mathbf{d}^\top \mathbf{d}$, $B_q = 2\mathbf{f}^\top \mathbf{d}$, and $C_q = \mathbf{f}^\top \mathbf{f} - \rho_z^2$. If the discriminant is negative, the overlap is zero. Otherwise, the two roots are clamped to $[0, 1]$ and sorted as $\lambda_1 \leq \lambda_2$. The segment length within zone z is

$$\text{ovl}(u \rightarrow v, z) = (\lambda_2 - \lambda_1)_+ \|\mathbf{p}_v - \mathbf{p}_u\|_2, \quad (2)$$

Aggregating over zones gives

$$R_{uv} = \sum_{z \in \mathcal{Z}^r} \text{ovl}(u \rightarrow v, z) + \frac{1}{2} \sum_{z \in \mathcal{Z}^c} \text{ovl}(u \rightarrow v, z), \quad H_{uv} = |\{z \in \mathcal{Z}^o : \text{ovl}(u \rightarrow v, z) > 0\}|. \quad (3)$$

Weak communication represents degradation in link reliability rather than a direct physical hazard. Its overlap coefficient is therefore set to one half of the unit coefficient assigned to the risk-zone term. The quantity H_{uv} records the number of obstacle regions intersected by segment (u, v) . The default scalar objective uses a high but finite obstacle penalty, so obstacle crossings are strongly penalized but not strictly prohibited.

Constructing $\mathbf{R}$ and $\mathbf{H}$ requires $\mathcal{O}(N^2(|\mathcal{Z}^r| + |\mathcal{Z}^c| + |\mathcal{Z}^o|))$ operations, while constructing Δ requires $\mathcal{O}(N^2)$ time. After preprocessing, retrieving a matrix value requires $\mathcal{O}(1)$ time per flight segment. A complete fitness evaluation then performs matrix lookups proportional to the number of decoded flight segments. This preprocessing avoids repeated geometric intersection calculations and supports comparison under a fixed fitness-evaluation budget.

2.2 Partial-completion service, risk-aware objective, and fitness evaluation

For an allocation vector $\mathbf{x}$, let $\mathcal{T}_i = \{j : x_j = i\}$ denote the tasks assigned to UAV i . Because $\mathbf{x}$ specifies task ownership but not execution order, a route-generation rule is required before fitness evaluation. The default decoder constructs the visit sequence π_i by nearest-neighbor ordering: starting from base b , UAV i repeatedly selects the unvisited assigned task with the smallest entry in the distance matrix Δ .

For the decoder-sensitivity analysis, the alternative risk-aware ordering selects the next task according to

$$\phi(u, j) = \frac{\Delta_{uj}}{s_D} + 1.5 \frac{R_{uj}}{s_R} + 2H_{uj} - 0.35 \frac{r_j}{s_r}, \quad j^* = \arg \min_{j \in \mathcal{T}_i^{\text{rem}}} \phi(u, j), \quad (4)$$

where u is the current node, $\mathcal{T}_i^{\text{rem}}$ is the set of remaining tasks assigned to UAV i , $s_D = \max\{S, 1\}$, $s_R = \max\{1, \max_{p,q} R_{pq}\}$, and $s_r = \max\{1, \max_j r_j\}$. The numerical coefficients in Eq. 4 are fixed across all algorithms and instances, whereas the scale factors s_D , s_R , and s_r are computed from each instance. If multiple tasks attain the same minimum score, the tie is resolved in favor of the smaller task index. Risk-stop and risk-skip use the same ordering and differ only in their treatment of an infeasible task. The prefix-stop policy represents a conservative pre-committed offline execution assumption and is not intended to model adaptive in-mission rescheduling. Its sensitivity is examined in Section 4.5.

Given the visiting order π_i , UAV i proceeds to serve the tasks sequentially while continuously updating its cumulative payload w_i and cumulative flight distance d_i . Specifically, the k -th task in the sequence, denoted by π_i^k , is accepted only if both the payload and endurance constraints remain satisfied after traversing the next flight segment:

$$w_i + q_{\pi_i^k} \leq Q_i \quad \text{and} \quad d_i + \Delta_{\text{prev}, \pi_i^k} \leq L_i. \quad (5)$$

If either condition is violated, the UAV terminates its service, and all subsequent tasks in π_i are marked as unserved under the default service-prefix policy. Consequently, the served tasks form a prefix $\mathcal{S}_i \subseteq \mathcal{T}_i$. This assumption may be conservative when an executable low-demand task follows an infeasible task; the decoder-sensitivity analysis therefore also evaluates skip-and-continue execution. Under skip-and-continue execution, the infeasible task is marked as unserved but the remaining ordered tasks continue to be examined. A skipped task does not update the previous node, cumulative payload, cumulative flight distance, or elapsed time. Consequently, the next task is evaluated from the last successfully served node, and only an accepted task updates the UAV state.

Compared with an all-or-nothing feasibility rule, prefix evaluation preserves credit for tasks completed before the first payload or endurance violation. Under an all-or-nothing rule, a small downstream violation can invalidate an entire assigned route even when several preceding tasks remain physically executable. Therefore, the present model scores the feasible prefix $\mathcal{S}_i$ and applies the missed-task penalty only to unserved tasks. This choice yields a more informative evaluation of the executable service. Because downstream tasks are not reconsidered under the default stopping policy, this evaluation remains conservative.

The overall objective function jointly accounts for travel efficiency, environmental exposure, time-window violations, workload balance, and task reward. For each flight segment served within the feasible prefix $\mathcal{S}_i$, the associated distance, risk overlap, and obstacle intersections are queried directly from the precomputed matrices $\{\Delta, \mathbf{R}, \mathbf{H}\}$. T_j denotes the arrival time at the served task j , calculated as the cumulative outbound distance of the assigned UAV along its feasible route divided by its constant speed. The individual cost and reward components are defined as

$$C_{\text{dis}} = \sum_i d_i, \quad (6)$$

$$C_{\text{risk}} = \sum_{\text{served flight segments}} R_{uv}, \quad (7)$$

$$C_{\text{time}} = \sum_{j \in \bigcup_i S_i} \max(0, T_j - \tau_j), \quad (8)$$

$$C_{\text{bal}} = \text{std}_i(d_i), \quad R_{\text{task}} = \sum_{j \in \bigcup_i S_i} r_j, \quad (9)$$

$$C_{\text{miss}} = N - \sum_i |S_i|, \quad P_{\text{obs}} = \kappa_o n_{\text{obs}}, \quad (10)$$

where $n_{\text{obs}} = \sum_{\text{served flight segments}} H_{uv}$ counts obstacle intersections over served flight segments. R_{uv} is defined in Eq. (3); C_{risk} includes risk-zone exposure and the weighted weak-communication overlap. For a task without a deadline, $\tau_j = \infty$, and its contribution to C_{time} is interpreted as zero. The balance term C_{bal} measures workload imbalance in terms of the outbound service distances of the UAVs.

Equations (2)-(3) and (5)-(10) are derived in this study from segment-circle geometry and the stated mission assumptions. Equation (1) follows the cited round-and-clip encoding practice.

Following the standard principle of decomposing multiple criteria into a scalar optimization objective^[37], the objective to be minimized is

$$F(\mathbf{x}) = \alpha C_{\text{dis}} + \beta C_{\text{risk}} + \gamma C_{\text{time}} + \delta C_{\text{bal}} - \eta R_{\text{task}} + \kappa C_{\text{miss}} + P_{\text{obs}}. \quad (11)$$

The fixed weights are $\alpha = 1$, $\beta = 2$, $\gamma = 3$, $\delta = 0.5$, $\eta = 1$, $\kappa = 80$, and $\kappa_o = 500$. The negative sign on the task reward R_{task} converts maximizing the reward into minimizing it. The risk and delay weights are set higher than the distance weight because risk exposure and delayed service represent mission degradation beyond travel length. The missed-task penalty is set higher than the reward of a single task to prevent a solver from sacrificing many tasks only to shorten distance or reduce exposure. The obstacle crossing penalty is a high-cost but non-locking term; a single crossing is more expensive than a moderate detour or reassignment, but the search space remains connected under the soft formulation. Because these terms have different numerical scales, Section 4.6 evaluates weight sensitivity and normalized-objective robustness. Fitness values obtained with different weight sets are interpreted as sensitivity outcomes.

In the experimental scenarios, it may not be possible to achieve 100% completion when the total task demand exceeds the total payload capacity of the UAV swarm. Therefore, a payload-only upper bound on the number of serviceable tasks is obtained by selecting tasks in ascending order of demand:

$$\rho_{\text{cap}} = \frac{1}{N} \max \left\{ k : \sum_{l=1}^k q_{(l)} \leq \sum_i Q_i \right\}, \quad (12)$$

where $q_{(1)} \leq q_{(2)} \leq \dots$ are the task demands sorted in ascending order. This theoretical ceiling ρ_{cap} represents the maximum fraction of tasks that can be served if payload were the sole constraint. However, in practice, endurance limitations, time-window requirements, and spatial penalties will degrade the realistically achievable service rate. Therefore, when a solver approaches ρ_{cap} , the remaining gap to this theoretical ceiling is more informative than the gap to the ideal 100% completion rate.

Algorithm 1 Evaluate(x) - partial-completion fitness

```

1:  $g \leftarrow g + 1$  ▷ one fitness evaluation
2:  $C_{\text{dis}}, C_{\text{risk}}, C_{\text{time}}, R_{\text{task}}, P_{\text{obs}} \leftarrow 0; n_{\text{done}} \leftarrow 0$ 
3: for each UAV  $i \in \mathcal{U}$  do
4:    $\mathcal{T}_i \leftarrow \{j : x_j = i\}; \pi_i \leftarrow \text{NearestNeighborTour}(b, \mathcal{T}_i)$ 
5:    $W_i \leftarrow 0; d_i \leftarrow 0; \text{prev} \leftarrow b$ 
6:   for  $j \in \pi_i$  in order do
7:      $s \leftarrow \Delta_{\text{prev}, j}$ 
8:     if  $W_i + q_j \leq Q_i$  and  $d_i + s \leq L_i$  then
9:        $P_{\text{obs}} \leftarrow P_{\text{obs}} + \kappa_o H_{\text{prev}, j}; C_{\text{risk}} \leftarrow C_{\text{risk}} + R_{\text{prev}, j}$ 
10:       $d_i \leftarrow d_i + s; W_i \leftarrow W_i + q_j; R_{\text{task}} \leftarrow R_{\text{task}} + r_j$ 
11:       $T_j \leftarrow d_i / v_i$ ; if  $T_j > \tau_j$  then  $C_{\text{time}} \leftarrow C_{\text{time}} + T_j - \tau_j$ 
12:       $n_{\text{done}} \leftarrow n_{\text{done}} + 1; \text{prev} \leftarrow j$ 
13:     else
14:       break ▷ stop the UAV; remaining tasks in  $\pi_i$  are unserved
15:     end if
16:   end for
17:    $C_{\text{dis}} \leftarrow C_{\text{dis}} + d_i$ ; store  $d_i$ 
18: end for
19:  $C_{\text{bal}} \leftarrow \text{std}_i(d_i); C_{\text{miss}} \leftarrow N - n_{\text{done}}$ 
20: return  $F(x)$  by Eq. (11)

```

Algorithm 1 summarizes the fitness evaluation. Each call to the objective function increments the global counter g by one. Thus, every evaluated candidate consumes one NFE, regardless of whether it is accepted. Route construction dominates the per-evaluation cost. In the worst case, its complexity is $O(\sum_i |\mathcal{T}_i|^2)$, and this sum is bounded by $O(N^2)$ because $\sum_i |\mathcal{T}_i|^2 \leq (\sum_i |\mathcal{T}_i|)^2 = N^2$. When the assignment is balanced, the cost is approximately $O(N^2/M)$. The subsequent feasibility sweep requires $O(N)$ operations. Since all algorithms rely on the same evaluator, NFE provides a common measure of objective-evaluation effort.

3. RA-IGWO FOR RISK-AWARE UAV TASK ALLOCATION

The decoded task-allocation problem produces a rugged fitness landscape in which objective values are jointly affected by payload violations, task omissions, route exposure, and workload imbalance. Continuous GWO updates do not always change the decoded assignment, which can promote stagnation on the many-to-one decoding plateaus. The implemented optimizer uses a quadratic convergence schedule, capacity-aware greedy initialization, OBL, allocation-level load-aware local refinement, and accept-only-improved population updates. All candidate fitness evaluations generated by population updates, OBL, and local refinement are counted against the same global budget G .

3.1 Standard GWO update and quadratic convergence schedule

In standard GWO^[21], the population is ranked by fitness, and the three best agents are denoted by z_α , z_β , and z_δ . For each leader $h \in \{\alpha, \beta, \delta\}$, two random vectors $r_{1,h}, r_{2,h} \sim \text{Unif}(0, 1)^N$ are sampled independently. The corresponding coefficient vectors are $A_h = 2ar_{1,h} - a$ and $C_h = 2r_{2,h}$. For a current search agent z , the candidate position directed by leader z_h is

$$X_h = z_h - A_h \odot |C_h \odot z_h - z| \quad (h \in \{\alpha, \beta, \delta\}), \quad \tilde{z} = \frac{1}{3} \sum_{h \in \{\alpha, \beta, \delta\}} X_h, \quad z' = \text{clip}(\tilde{z}, \ell, u). \quad (13)$$

where $\odot$ and $|\cdot|$ denote elementwise multiplication and elementwise absolute value, respectively. The coefficient pairs (A_α, C_α) , (A_β, C_β) , and (A_δ, C_δ) are generated independently, matching the implementation in which each

leader call draws new random vectors. The final clipping operation implements the boundary handling used by all GWO-based methods.

Since each component of $r_{1,h}$ lies in $[0, 1]$, every component of A_h lies in $[-a, a]$. Consequently, $|A_{h,k}| > 1$, which permits a candidate to move away from a leader, is possible only when $a > 1$. Conversely, $|A_{h,k}| < 1$ favors movement toward the corresponding leader.

In standard GWO, a decreases linearly from 2 to 0^[21]. Using normalized evaluation progress $p = g/G$, the linear schedule is $a_{\text{lin}}(p) = 2(1 - p)$. It reaches $a = 1$ at $p = 0.5$, after which the condition $|A_{h,k}| > 1$ is no longer possible. This threshold does not eliminate stochastic movement because c_h remains random; it only marks the end of the interval in which the magnitude of A_h can exceed one. RA-IGWO replaces the linear decay with the quadratic schedule

$$a(g) = 2 \left[1 - (g/G)^2 \right], \quad (14)$$

while retaining the three-leader update in Eq. (13). For $0 < p < 1$, $a(p) = 2(1 - p^2) > a_{\text{lin}}(p)$, so the quadratic schedule retains a larger coefficient range at the same normalized NFE progress. Moreover, $a(p) > 1$ is equivalent to $p < 1/\sqrt{2} \approx 0.707$. Therefore, the interval during which $|A_{h,k}| > 1$ is possible increases from $0.5G$ to approximately $0.707G$, an absolute increase of $0.207G$ and a relative increase of $\sqrt{2}-1 \approx 41.4\%$. The magnitude of its slope increases with p , while the convergence factor itself reaches zero at the terminal budget G . This schedule is designed to retain exploration-capable updates for a larger fraction of the NFE budget.

3.2 Capacity-aware greedy initialization, OBL, and local refinement

The remaining components are designed around the decoded assignment structure. During initialization, $\lfloor P/3 \rfloor$ agents are generated using a capacity-aware greedy heuristic. The first greedy pass assigns tasks in descending order of demand; the subsequent passes employ randomized task sequences to maintain population diversity. For each task, the heuristic selects the feasible UAV with the largest remaining payload capacity. If no UAV has sufficient capacity to meet the task requirements, the task will be assigned to the UAV with the largest remaining capacity, thereby providing a near-feasible load distribution for subsequent evolutionary optimization. The obtained integer allocation is then mapped back to the continuous search space $[\ell, u]^N$, and a small amount of uniform perturbation is added. This process provides feasible or near-feasible load distributions for a subset of individuals in the population while ensuring that the initial individuals are not completely identical.

OBL^[29] is used to expand the sampling area of the continuous search space. The opposite solution of a proxy z is defined as

$$\tilde{z} = \ell + u - z. \quad (15)$$

After evaluating the opposite candidates at initialization, the best P agents are retained from the merged original and opposite populations. During the early search stage, when $g/G < 0.4$, generalized OBL is triggered with probability p_j . Its coordinate-wise lower and upper bounds are obtained from the current population, and every reflected candidate is clipped to $[\ell, u]^N$. Because UAV labels have no inherent ordinal meaning, OBL is performed in the common continuous encoding space and evaluated after decoding.

The alpha wolf is further refined by first-improvement search in the decoded integer space. The search samples three neighborhood types: load-aware transfer, reassignment, and two-task swapping. The load-aware transfer

moves a task from the most capacity-constrained UAV to a UAV with larger remaining capacity to reduce service loss caused by payload limits, while the other operations maintain neighborhood diversity. Each sampled local-refinement candidate is evaluated by Algorithm 1 and consumes one NFE regardless of whether it improves the incumbent solution. This rule prevents local search from receiving an uncounted evaluation advantage over non-memetic baselines.

3.3 Overall RA-IGWO procedure and evaluation-budget accounting

Algorithm 2 summarizes the complete optimizer. Greedy seeding, OBL refresh, GWO position updates, and assignment-level local refinement all follow the same NFE accounting rule. After local refinement, the alpha solution is written back to the population before the leaders are updated again; otherwise, the refined assignment could be lost during the next population ranking.

Algorithm 2 RA-IGWO with embedded greedy seeding and local refinement

```

1: Input: evaluation budget  $G$ , population size  $P$ , jump rate  $p_j$ 
2:  $g \leftarrow 0$ ; initialize  $\lfloor P/3 \rfloor$  greedy agents and  $P - \lfloor P/3 \rfloor$  random agents in  $[\ell, u]^N$ 
3: evaluate all agents by Algorithm 1; evaluate opposite agents by Eq. (15); retain the best  $P$  agents and update leaders
4: while  $g < G$  do
5:   set  $a \leftarrow 2[1 - (g/G)^2]$  and update the population by the GWO move in Eq. (13), accepting only improved candidates
6:   if  $g/G < 0.4$  and  $\text{rand}() < p_j$  then
7:     perform generalized OBL using current population bounds; evaluate opposite agents and retain the best  $P$  agents
8:   end if
9:   update leaders; decode the current alpha wolf as  $x^*$  and record its population index  $p_\alpha$ 
10:  refine  $x^*$  by first-improvement assignment moves for at most  $\max\{12, \lfloor N/3 \rfloor\}$  trials while  $g < G$ 
       $\triangleright$  each evaluated local candidate consumes one NFE regardless of acceptance
11:  write the refined leader back to the population,  $z_{p_\alpha} \leftarrow \text{lift}(x^*)$ ,  $F_{p_\alpha} \leftarrow F^*$ , and update leaders again
12: end while
13: return  $\text{dec}(z_\alpha), F_\alpha$ 

```

4. SIMULATION RESULTS AND ROBUSTNESS ANALYSIS

Experiments use Python 3.12.0 on an Intel Core i9-13900HX workstation with 16GB RAM and no algorithm-level parallelism. All plots and route visualizations are author-generated from the reported outputs using Python, NumPy, and Matplotlib.

RA-IGWO is compared with GA, PSO, GWO, I-GWO, HPSOGWO, and a discrete adaptive large neighborhood search benchmark (D-ALNS). Representative and hard-obstacle tests use $(M, N) = (5, 30), (8, 50),$ and $(10, 80)$; multi-instance tests use $(5, 20), (8, 50),$ and $(10, 100)$, denoted Small-20, Medium-50, and Large-100, where the suffix denotes tasks. Each multi-instance setting contains 20 environments. Population methods share the continuous encoding, round-and-clip decoder, boundary clipping, and $P = 30$; D-ALNS searches integer ownership directly. All methods use the same evaluator, paired environments, and $G = 4000$ NFEs, with runtime recorded.

4.1 Experimental setup and fairness protocol

All methods follow one evaluation protocol. Table 2 lists all settings. Every RA-IGWO candidate produced by a GWO update, OBL refresh, or local refinement consumes one NFE; the final diagnostic call reports metrics only. Reference configurations are used for GA [9,10], PSO [38,39,11], GWO [21], I-GWO [24], HPSOGWO [31,32], and UAV multi-task ALNS [40]. D-ALNS adapts the latter framework to ownership-only decisions. Its random, worst-contribution, overload-oriented, and risk-oriented removal operators are paired with greedy, regret-2, and capacity-balanced repair, adaptive operator weights, and simulated-annealing acceptance. Each repaired assignment consumes one NFE. GWO uses $a : 2 \rightarrow 0$. PSO uses $w = 0.70, c_1 = c_2 = 1.50$, close to stability-oriented literature coefficients rather than a claimed optimum, while GA uses the predeclared $p_c = 0.80, p_m = 0.10$.

Table 2. Algorithm and environment parameters

Component	Parameter	Value	Parameter	Value
All methods	NFE budget G	4,000	Independent instances/seeds	20/5
Population methods	Pop size P	30	Boundary handling	Clipping
GA	Crossover prob.	0.80	Mutation prob.	0.10
PSO	Inertia w	0.70	Accel. c_1, c_2	1.5, 1.5
HPSOGWO	Coeff. (w, c_1, c_2, c_3)	0.5, 0.5, 0.5, 0.5	–	–
D-ALNS	Destroy fractions	0.10, 0.20, 0.30	Reaction/cooling	0.20/0.999
RA-IGWO	Jump rate p_j	0.30	Greedy seed count	$\lfloor P/3 \rfloor$
Environment	Area side s	100	Task position	$\text{Unif}(5, s - 5)^2$
	UAV speed v_i	$\text{Unif}(0.8, 1.5)$	UAV payload Q_i	$\text{Unif}(8.0, 14.0)$
	UAV range L_i	$\text{Unif}(280, 420)$	Task demand q_j	$\text{Unif}(1.0, 3.0)$
	Task reward r_j	$\text{Unif}(1.0, 10.0)$	Tasks with time windows	$\max(1, \lfloor N/3 \rfloor)$
	Deadline τ_j	$\text{Unif}(80, 200)$	Obstacle zones	2
	Weak-comm. zones	2	Zone radius (risk/comm.)	$\text{Unif}(8.0, 14.0)$
	Zone radius (obstacle)	$\text{Unif}(6.0, 10.0)$	–	–
Fitness weights	$\alpha, \beta, \gamma, \delta, \eta$	1, 2, 3, 0.5, 1	κ, κ_o	80, 500

$\text{Unif}(a, b)^2$ denotes two independently sampled Cartesian coordinates, so that each task position is sampled uniformly over $[a, b] \times [a, b]$.

Table 3. Parameter sensitivity of GA, PSO, and GWO on medium-scale instances under the shared 4000-NFE budget

Algorithm	Parameter setting	Runs	Fitness mean $\pm$ std	Completion
GA	$p_c = 0.80, p_m = 0.10$ default	10	$3,128.97 \pm 539.84$	68.6%
GA	$p_c = 0.70, p_m = 0.10$	10	$3,107.33 \pm 648.65$	68.4%
GA	$p_c = 0.90, p_m = 0.10$	10	$2,863.13 \pm 334.51$	71.6%
GA	$p_c = 0.80, p_m = 0.05$	10	$2,198.69 \pm 268.28$	74.4%
GA	$p_c = 0.80, p_m = 0.20$	10	$3,500.42 \pm 759.65$	69.0%
PSO	$w = 0.70, c_1 = c_2 = 1.50$ default	10	$2,047.92 \pm 173.35$	74.2%
PSO	$w = 0.40, c_1 = c_2 = 1.50$	10	$2,150.25 \pm 185.45$	73.6%
PSO	$w = 0.70, c_1 = c_2 = 1.20$	10	$2,019.49 \pm 228.44$	76.4%
PSO	$w = 0.70, c_1 = c_2 = 1.80$	10	$2,225.42 \pm 298.51$	71.8%
PSO	$w = 0.90, c_1 = c_2 = 1.50$	10	$2,436.36 \pm 209.95$	68.6%
GWO	$a : 2.00 \rightarrow 0.00$ default	10	$2,247.38 \pm 244.59$	70.0%
GWO	$a : 1.50 \rightarrow 0.00$	10	$2,055.63 \pm 219.43$	73.8%
GWO	$a : 2.50 \rightarrow 0.00$	10	$2,404.74 \pm 526.87$	69.6%
GWO	$a : 2.00 \rightarrow 0.50$	10	$2,414.42 \pm 303.24$	67.8%

With $P = 30 < G = 4,000$, initial population evaluations are counted first, and every later evaluator call checks the remaining budget. Population and OBL batches are truncated at exhaustion, and local refinement stops at $g = G$. Unchanged local moves are discarded before evaluation; every changed candidate consumes one NFE regardless of acceptance. D-ALNS removal, repair, insertion scoring, and weight updates use assignment data and precomputed matrices; each repaired assignment invokes the full evaluator once. Thus, all seven methods terminate at $g = 4,000$ without exceeding the budget.

Runtime covers optimizer construction and search. Environment generation, geometric preprocessing, output, visualization, and the post-run diagnostic call are excluded, yielding implementation-level comparisons in a common environment.

Table 4. Confirmation using the strongest observed GA, PSO, and GWO settings on independent medium-scale instances

Algorithm	Confirmation setting	Runs	Fitness mean $\pm$ std	Completion	FDR q
GA	$p_c = 0.80, p_m = 0.05$	100	$2,305.16 \pm 407.02$	73.0%	< 0.0001
PSO	$w = 0.70, c_1 = c_2 = 1.20$	100	$2,177.37 \pm 480.94$	75.0%	< 0.0001
GWO	$a : 1.50 \rightarrow 0.00$	100	$2,293.82 \pm 465.06$	70.9%	< 0.0001
I-GWO	Reference implementation	100	$2,500.96 \pm 483.80$	68.4%	< 0.0001
HPSOGWO	Reference implementation	100	$2,453.13 \pm 485.13$	70.1%	< 0.0001
RA-IGWO	Proposed configuration	100	$1,688.91 \pm 374.51$	83.6%	–

Each row comprises 20 independent environments and five seeds per environment. FDR-adjusted q -values compare each baseline with RA-IGWO. Bold indicates the lowest mean fitness among all compared algorithms.

Table 5. Representative single-run performance under a 4000-NFE budget

Scenario	Algorithm	Fitness	Distance	Risk	#Tardy	Completion	Cap. ρ_{cap}	Balance	Time (s)
Small	GA	1,835.79	658.7	110.6	0	76.7%	93.3%	51.49	0.64
Small	PSO	1,528.85	589.7	122.4	0	86.7%		38.30	0.56
Small	GWO	1,152.92	534.4	43.2	0	73.3%		36.83	0.62
Small	I-GWO	1,072.82	594.8	99.8	0	83.3%		51.59	0.66
Small	HPSOGWO	1,374.89	533.6	109.2	0	70.0%		59.27	0.67
Small	RA-IGWO	893.03	569.1	108.8	0	90.0%		38.72	0.65
Medium	GA	2,470.62	835.1	145.2	0	62.0%	88.0%	47.08	0.96
Medium	PSO	2,801.34	971.4	127.2	0	80.0%		40.81	0.89
Medium	GWO	2,065.81	914.6	119.6	0	72.0%		28.78	0.94
Medium	I-GWO	2,844.55	608.9	95.0	0	46.0%		43.73	0.99
Medium	HPSOGWO	2,582.86	722.2	166.4	0	58.0%		54.18	0.99
Medium	RA-IGWO	1,987.10	780.4	73.6	0	80.0%		31.20	0.95
Large	GA	4,549.23	1,114.6	510.7	1	58.8%	80.0%	36.44	1.41
Large	PSO	3,903.63	1,038.3	370.1	0	62.5%		30.18	1.40
Large	GWO	4,344.36	1,164.5	427.0	2	60.0%		39.10	1.39
Large	I-GWO	5,162.62	1,155.8	562.8	0	58.8%		18.36	1.49
Large	HPSOGWO	4,815.38	1,032.7	535.2	0	53.8%		42.17	1.48
Large	RA-IGWO	3,228.84	1,053.0	357.7	0	72.5%		27.51	1.46

Bold indicates the lowest fitness and the highest or tied-highest completion rate per scenario. Cap. ρ_{cap} : payload-based completion ceiling (Eq. 12); '# Tardy': number of tasks missing their deadline; 'Balance': standard deviation of per-UAV distances (c_{bal} in Eq. 11).

Table 3 examines GA, PSO, and GWO parameters under the common population and NFE settings; it is a sensitivity analysis, not independent optimization. Mean fitness spans 2,198.69–3,500.42 for GA, 2,019.49–2,436.36 for PSO, and 2,055.63–2,414.42 for GWO, so the predeclared settings are treated as reproducible rather than optimal. The strongest observed configurations are then tested on 20 new environments with five paired seeds. I-GWO and HPSOGWO retain their reference implementations. In Table 4, RA-IGWO obtains mean fitness 1,688.91 vs. 2,305.16, 2,177.37, 2,293.82, 2,500.96, and 2,453.13 for GA, PSO, GWO, I-GWO, and HPSOGWO, respectively, with all FDR-adjusted $q < 0.0001$.

4.2 Representative main comparison

Table 5 gives one reproducible 4000-NFE run to illustrate distance, risk, completion, balance, and runtime jointly.

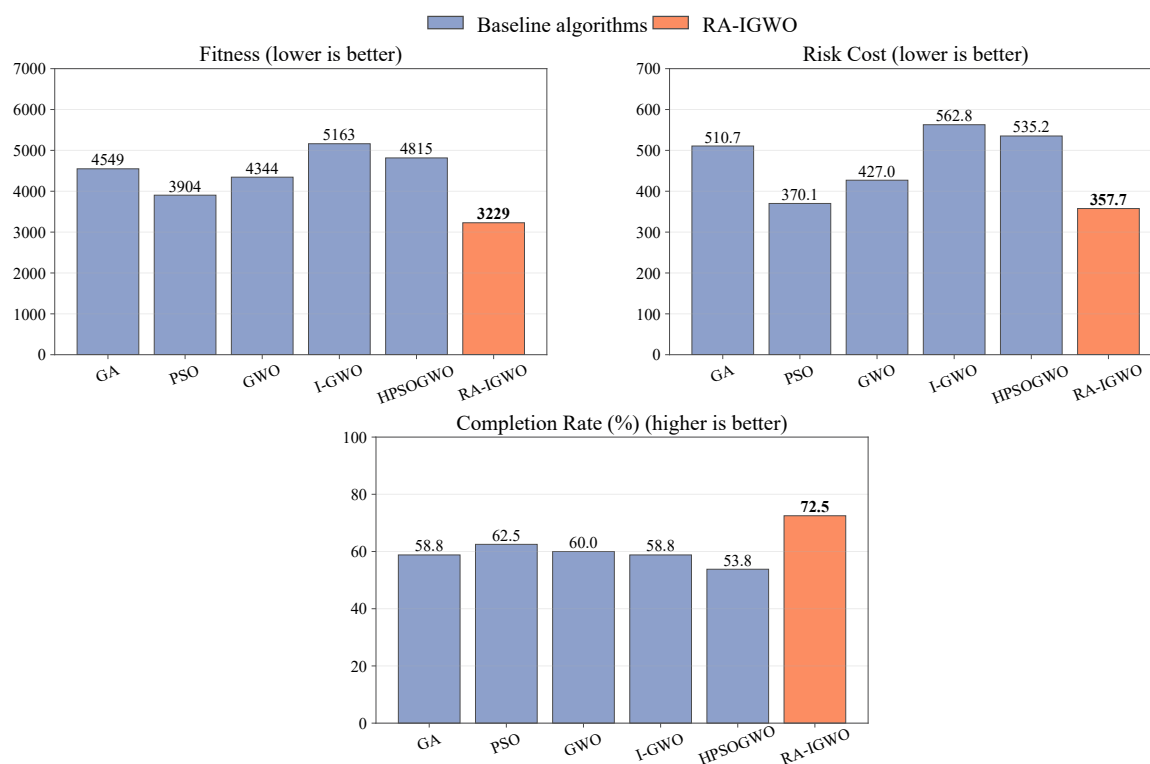


Figure 1. Large-scale representative metric comparison under the shared 4000-NFE budget. Blue bars denote baseline algorithms, and orange bars denote RA-IGWO. Bold value labels denote the best result in each panel, namely the lowest fitness and risk cost and the highest completion rate.

RA-IGWO has the lowest fitness and highest or tied-highest completion in all three representative runs. Figure 1 visualizes the large-scale metric decomposition. Because one run is descriptive rather than inferential, robustness is assessed below using independent instances, confidence intervals, effect sizes, and sensitivity tests.

4.3 Convergence behavior

Figure 2 plots best-so-far fitness against NFE for all three representative scales. Capacity-aware seeding structures initial allocations, OBL broadens sampling, and the quadratic schedule prolongs the interval with $a > 1$ from $g/G < 1/2$ to $g/G < 1/\sqrt{2}$. Local search changes decoded ownership directly and can repair capacity-related service loss that continuous moves leave unchanged. The curves describe the combined implementation; separate component effects and interactions are evaluated in Section 4.8.

4.4 Obstacle-model validation

Because a finite penalty can permit crossings, Table 6 reports mean intersections over 20 environments and five seeds per method and instance. RA-IGWO ranks first in the medium and large cases and second in the small case. These results quantify penalized soft-obstacle behavior and must not be interpreted as strict avoidance.

Table 6. Obstacle-intersection statistics under the default soft-obstacle formulation

Scenario	GA	PSO	GWO	I-GWO	HPSOGWO	RA-IGWO
Small-20	0.32	0.37	0.31	0.26	0.34	0.29
Medium-50	0.72	0.10	0.15	0.06	0.15	0.03
Large-100	1.36	0.12	0.35	0.33	0.30	0.10

Entries are means over 100 runs; bold marks the lowest value in each scenario.

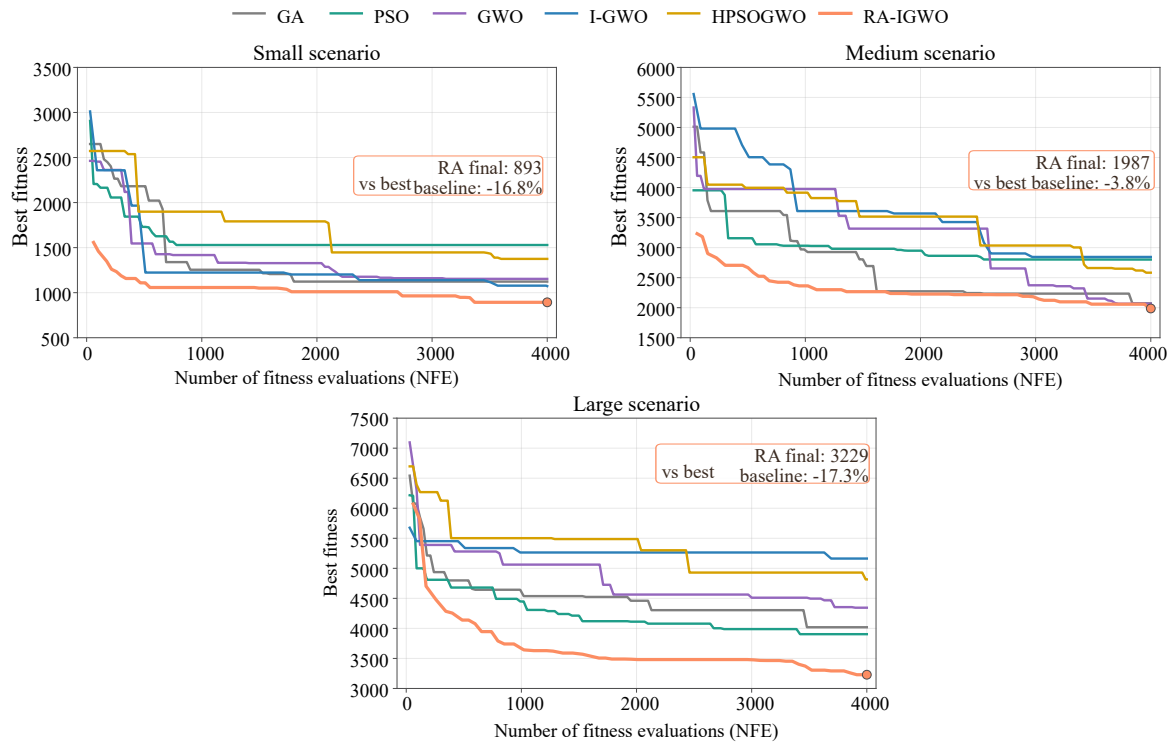


Figure 2. Convergence curves under the shared NFE budget for small, medium, and large-scale scenarios.

Table 7 enforces hard obstacles. A task j is rejected before state update when the prospective segment from the last served node u satisfies $H_{uj} > 0$. Under the tested prefix-stop policy, this terminates only that UAV route and places the blocked and downstream tasks in C_{miss} without assigning infinite fitness to the complete candidate; already served tasks remain valid. With skip-and-continue, which is not used in this table, only the blocked task would be skipped from an unchanged previous node. All reported crossings are therefore zero. RA-IGWO has the lowest mean fitness and highest completion at every scale, so its population-baseline advantage is not attributable to permissive crossings.

Table 8 varies κ_o from 100 to 2,000 over 20 medium-scale environments. RA-IGWO remains below the strongest population baseline, PSO, at every setting, with all FDR-adjusted comparisons significant. Its mean crossings fall from 0.66 to 0.24 while completion remains within 81.9%–82.8%, indicating stable ranking across the tested finite penalties.

4.5 Decoder sensitivity

Table 9 applies four decoders to all six population methods over 20 medium-scale instances and five seeds. Risk-aware ordering reduces RA-IGWO fitness from 1,766.47 to 1,600.78 and eliminates obstacle crossings. For RA-IGWO, skip-and-continue increases completion from 84.8% to 85.3% under nearest-neighbor ordering and leaves completion unchanged at 85.3% under risk-aware ordering. For PSO, completion changes slightly from 78.1% to 78.0% and from 80.7% to 80.3%, respectively. The completion effect of skipping is small and method-dependent. PSO is the strongest population-based baseline for each decoder, while RA-IGWO retains lower fitness than all five population-based baselines under every decoder, with all 20 joint FDR-adjusted comparisons significant ($q \leq 0.0003$). Thus, the ranking is not confined to nearest-neighbor prefix stopping.

Table 7. Hard-obstacle validation for the six population-based algorithms

Scenario	Algorithm	Fitness mean	Completion	Crossings	Runtime (s)
Small	GA	1,192.60	72.6%	0.00	0.686
Small	PSO	1,022.73	79.1%	0.00	0.646
Small	GWO	1,023.95	77.9%	0.00	0.721
Small	I-GWO	1,036.32	78.0%	0.00	0.754
Small	HPSOGWO	1,048.90	77.7%	0.00	0.741
Small	RA-IGWO	918.47	82.0%	0.00	0.724
Medium	GA	2,293.60	67.0%	0.00	1.026
Medium	PSO	1,806.23	77.8%	0.00	0.998
Medium	GWO	2,009.40	73.1%	0.00	1.071
Medium	I-GWO	2,103.02	71.5%	0.00	1.099
Medium	HPSOGWO	2,082.71	71.9%	0.00	1.087
Medium	RA-IGWO	1,559.83	83.2%	0.00	1.090
Large	GA	4,431.34	52.1%	0.00	1.581
Large	PSO	3,680.88	61.7%	0.00	1.539
Large	GWO	4,049.59	56.8%	0.00	1.619
Large	I-GWO	4,224.20	54.4%	0.00	1.658
Large	HPSOGWO	4,195.09	55.2%	0.00	1.637
Large	RA-IGWO	3,264.90	66.4%	0.00	1.685

Bold indicates the lowest mean fitness and the highest completion rate within each scenario.

Table 8. Cross-algorithm obstacle-penalty sensitivity on medium-scale instances

κ_o	PSO fitness	RA-IGWO fitness	RA completion	RA crossings	FDR q
100	1,921.20	1,679.54	82.8%	0.66	< 0.0001
250	2,009.85	1,716.62	82.4%	0.35	< 0.0001
500	2,059.88	1,809.21	82.4%	0.32	0.0003
1,000	2,249.11	1,964.54	81.9%	0.30	< 0.0001
2,000	2,520.28	2,121.70	82.2%	0.24	< 0.0001

PSO is shown because it has the lowest fitness among the five population-based baselines at every tested penalty. Each entry summarizes 100 runs. Bold indicates the lowest RA-IGWO mean fitness among the tested obstacle-penalty values.

Table 9. Cross-algorithm decoder sensitivity on medium-scale instances

Decoder	PSO fitness	RA fitness	PSO comp.	RA comp.	RA crossings	FDR q
NN-stop	2,124.53	1,766.47	78.1%	84.8%	0.42	< 0.0001
Risk-stop	1,830.91	1,600.78	80.7%	85.3%	0.00	< 0.0001
NN-skip	2,122.80	1,757.47	78.0%	85.3%	0.36	< 0.0001
Risk-skip	1,869.79	1,597.60	80.3%	85.3%	0.00	< 0.0001

PSO is the strongest of the five baselines for each decoder. The reported q -value is for PSO versus RA-IGWO; comparisons with the other four baselines are also significant. Bold indicates the lower mean fitness between PSO and RA-IGWO under each decoder.

4.6 Objective-scale, weight, and normalization sensitivity

Objective-scale robustness is examined through empirical term magnitudes, common weight profiles, and normalization. Across 2,000 random medium-scale assignments, median absolute weighted contributions for distance, risk, delay, imbalance, reward, missed tasks, and obstacles are 1,051.25, 727.50, 84.85, 19.82, 196.74, 1,040.00, and 3,000.00. The dominant missed-task and obstacle terms motivate the checks below. Because each profile defines a different scalar objective, raw fitness is compared within a common profile.

Table 10 reports objective-weight sensitivity. The three-profile comparison and separate five-level β sweep used distinct Monte Carlo batches. Each used 20 independently generated medium-scale instances with five algorithm seeds per instance, giving 100 runs per setting, but the instance and algorithm-seed blocks differed be-

Table 10. Cross-algorithm objective-weight profiles on medium-scale instances

Profile	PSO fitness	RA fitness	RA completion	RA risk	RA crossings	FDR q
Completion-oriented	1,901.72	1,658.34	84.0%	203.70	0.54	0.0020
Balanced	2,154.68	1,874.05	82.9%	141.57	0.53	< 0.0001
Risk-oriented	2,904.99	2,426.16	77.2%	80.71	0.48	< 0.0001

The weight vectors $(\alpha, \beta, \gamma, \delta, \eta, \kappa, \kappa_0)$ are (1, 0.5, 3, 0.5, 1, 80, 500), (1, 2, 3, 0.5, 1, 80, 500), and (1, 8, 3, 0.5, 1, 80, 500) for the completion-oriented, balanced, and risk-oriented profiles, respectively. PSO is the strongest population-based baseline under each profile. Each row summarizes 100 runs; all comparisons with the other four baselines are also significant after FDR correction. Bold indicates the lower mean fitness between PSO and RA-IGWO within each objective-weight profile.

tween the two experiments. Although the $\beta = 0.5$ and $\beta = 8$ weight vectors are identical in the two experiments, their results use different instance and algorithm-seed sets and need not coincide numerically. Both show the same completion-risk trade-off: Table 10 reports a risk reduction from 203.70 to 80.71 and a completion reduction from 84.0% to 77.2%, but the separate sweep reports changes from 206.31 to 83.40 and from 85.1% to 78.6%. RA-IGWO remains lower in fitness than all five population-based baselines under every common profile.

For normalized-objective testing, distance, risk, delay, balance, reward, missed-task, and obstacle terms are divided by scales s_N , $N \max_{uv} R_{uv}$, the largest finite deadline, s , $\sum_j r_j$, N , and $N \max\{|\mathcal{Z}^o|, 1\}$, respectively; the original weights and signs are retained, and each scale is lower-bounded by one. The delay scale is

$$s_{\text{time}} = \max\left\{1, \max_{j: \tau_j < \infty} \tau_j\right\}, \quad (16)$$

where tasks with infinite deadlines are excluded and contribute zero delay. If no finite deadline exists, the inner maximum is defined as zero, giving $s_{\text{time}} = 1$ and an identically zero delay term. All instance-specific scales are shared by the compared methods. In Table 11, RA-IGWO is below all population baselines after FDR correction, while its difference from D-ALNS is not significant ($q = 0.4524$), indicating comparable normalized-objective performance.

Table 11. All-algorithm comparison under the normalized objective

Algorithm	Fitness mean $\pm$ std	Completion	Crossings	Runtime (s)
GA	24.31 $\pm$ 6.08	75.2%	0.82	1.191
PSO	13.61 $\pm$ 4.61	84.2%	0.22	1.142
GWO	18.62 $\pm$ 5.67	79.2%	0.35	1.230
I-GWO	19.14 $\pm$ 4.72	77.9%	0.23	1.273
HPSOGWO	18.38 $\pm$ 5.52	78.7%	0.23	1.262
D-ALNS	11.49 $\pm$ 4.54	86.2%	0.14	4.316
RA-IGWO	11.35 $\pm$ 5.24	86.5%	0.18	1.194

Bold indicates the best value in each metric column: the lowest fitness, number of crossings, and runtime and the highest completion rate.

4.7 Multi-instance generalization and statistical evidence

Each Small-20, Medium-50, and Large-100 setting uses 20 independently generated instances and five seeds per method and instance. The environment is the statistical unit: seeds are averaged within an instance before paired inference. For each environment and seed index, the same integer seed is reset before every algorithm. Table 12 reports the strongest population comparator, D-ALNS, and RA-IGWO; Table 13 covers all seven methods with Benjamini-Hochberg correction jointly applied across 18 comparisons.

Table 12. Multi-instance generalization summary

Scenario	Algorithm	Fitness mean	Completion	Crossings	Runtime (s)
Small-20	I-GWO	790.56	88.8%	0.26	0.735
Small-20	D-ALNS	1,042.09	98.2%	1.02	1.839
Small-20	RA-IGWO	709.30	94.4%	0.29	0.671
Medium-50	PSO	1,970.25	76.6%	0.10	1.202
Medium-50	D-ALNS	1,695.54	82.1%	0.09	4.474
Medium-50	RA-IGWO	1,608.30	83.0%	0.03	1.178
Large-100	PSO	5,092.63	53.7%	0.12	2.097
Large-100	D-ALNS	4,642.23	55.0%	0.00	11.233
Large-100	RA-IGWO	4,564.23	56.8%	0.10	2.287

Bold indicates the best value for each metric within each scenario: the lowest fitness, number of crossings, and runtime and the highest completion rate.

Table 13. FDR-adjusted paired Wilcoxon evidence for RA-IGWO versus all baselines

Scenario	Baseline	ΔF [95%CI]	FDR q	d_z	RA better
Small-20	GA	244.76 [203.58, 292.97]	< 0.0001	2.359	100%
Small-20	PSO	119.34 [82.06, 164.96]	< 0.0001	1.218	100%
Small-20	GWO	104.20 [86.11, 124.42]	< 0.0001	2.335	100%
Small-20	I-GWO	81.26 [60.91, 102.37]	< 0.0001	1.678	100%
Small-20	HPSOGWO	123.72 [103.50, 147.34]	< 0.0001	2.410	100%
Small-20	D-ALNS	332.78 [188.08, 499.71]	0.0010	0.902	70%
Medium-50	GA	1,272.26 [1160.22, 1385.95]	< 0.0001	4.741	100%
Medium-50	PSO	361.95 [311.67, 411.38]	< 0.0001	3.074	100%
Medium-50	GWO	682.30 [604.85, 770.83]	< 0.0001	3.464	100%
Medium-50	I-GWO	808.74 [738.66, 892.07]	< 0.0001	4.541	100%
Medium-50	HPSOGWO	731.28 [652.85, 819.02]	< 0.0001	3.768	100%
Medium-50	D-ALNS	87.23 [19.03, 178.47]	0.0121	0.460	65%
Large-100	GA	1,916.12 [1652.70, 2183.74]	< 0.0001	3.062	100%
Large-100	PSO	528.40 [463.71, 595.78]	< 0.0001	3.420	100%
Large-100	GWO	1,098.73 [935.24, 1258.21]	< 0.0001	2.874	100%
Large-100	I-GWO	1,248.65 [1096.16, 1405.13]	< 0.0001	3.392	100%
Large-100	HPSOGWO	1,169.16 [1006.94, 1339.54]	< 0.0001	2.938	100%
Large-100	D-ALNS	78.01 [−0.75, 143.83]	0.0113	0.462	75%

For instance k , $\Delta F_k = \bar{F}_{\text{baseline},k} - \bar{F}_{\text{RA-IGWO},k}$ after averaging five seeds. The percentile-bootstrap CI uses the 20 paired differences, and paired Cohen's $d_z = \Delta F / s_{\Delta F}$, where $s_{\Delta F}$ is their sample standard deviation. Positive values favor RA-IGWO.

Relative to D-ALNS, RA-IGWO reduces mean fitness by 332.78, 87.23, and 78.01 at the three scales, with significant FDR-adjusted signed-rank tests. However, for Large-100, the bootstrap CI [−0.75, 143.83] includes zero, so interval and rank evidence differ, and the large-scale advantage should be interpreted cautiously. D-ALNS has the highest small-case completion and the fewest large-case crossings. RA-IGWO has fewer crossings at the other scales and is 2.7–4.9 times faster. Its small-case fitness advantage mainly reflects lower obstacle-intersection penalties despite lower completion. Therefore, the results indicate that lower default-objective values, not universal dominance across completion, crossings, and runtime.

Table 14. Matched robustness comparison with the discrete D-ALNS baseline

Setting	D-ALNS fitness	RA-IGWO fitness	D-ALNS comp.	RA-IGWO comp.	ΔF	FDR q
Hard, Small-30	879.85 ± 137.47	918.47 ± 145.77	82.7%	82.0%	−38.62	< 0.0001
Hard, Medium-50	1,510.12 ± 219.52	1,559.83 ± 215.39	83.6%	83.2%	−49.71	< 0.0001
Hard, Large-80	3,321.16 ± 284.78	3,264.90 ± 297.53	65.1%	66.4%	56.26	0.0064
Normalized, Medium-50	11.49 ± 4.54	11.35 ± 5.24	86.2%	86.5%	0.14	0.4524

$\Delta F = F_{\text{D-ALNS}} - F_{\text{RA-IGWO}}$; positive values favor RA-IGWO. Each row contains 20 independent instances and five seeds per method and instance.

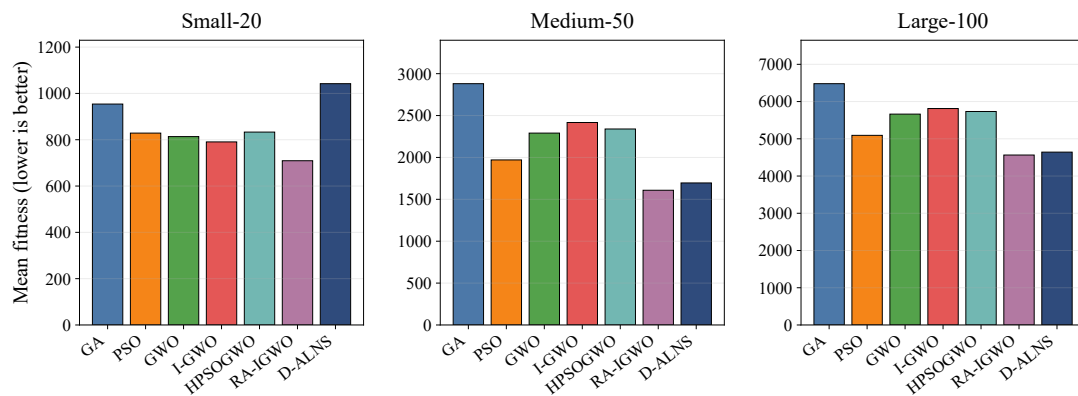


Figure 3. Mean fitness over independently generated spatial instances.

Table 14 shows scale-dependent hard-obstacle results: D-ALNS is significantly better for Small-30 and Medium-50, but RA-IGWO is better for Large-80. Under normalization they are statistically indistinguishable. Thus, RA-IGWO remains superior to the five population baselines under hard obstacles but not uniformly to the discrete baseline. Separate medium-scale D-ALNS sensitivity checks vary destroy fraction, reaction factor, and cooling rate, and none differs significantly from the predeclared configuration after FDR correction. Figure 3 displays all seven methods across the three multi-instance scales.

4.8 Component ablation

Table 15 uses 20 medium instances and isolates each component. Removing local refinement or accept-only updates raises fitness by 740.46 and 480.55, respectively, with FDR significance. Greedy initialization has mixed evidence: $q = 0.0551$, but its bootstrap CI excludes zero. OBL has no significant independent effect ($q = 0.0826$), with a CI containing zero. Removing the quadratic schedule yields a small significant improvement ($\Delta F = -23.90$, 95%CI $[-45.64, -1.44]$, $q = 0.0399$). Accordingly, it is retained only as the pre-specified exploration policy.

Table 15. Multi-instance leave-one-component-out and elitist-update ablation

Variant	Fitness mean $\pm$ std	ΔF [95%CI]	FDR q	Completion
Full	1,634.84 $\pm$ 262.99	—	—	83.7%
Without greedy	1,601.65 $\pm$ 258.83	−33.19 [−61.97, −6.21]	0.0551	84.1%
Without OBL	1,620.45 $\pm$ 280.45	−14.39 [−46.30, 22.90]	0.0826	84.2%
Without quadratic	1,610.94 $\pm$ 265.25	−23.90 [−45.64, −1.44]	0.0399	84.0%
Without local	2,375.30 $\pm$ 281.44	740.46 [672.94, 804.84]	< 0.0001	69.7%
Without elitist update	2,115.39 $\pm$ 523.32	480.55 [387.45, 581.25]	< 0.0001	76.4%

$\Delta F = F_{\text{variant}} - F_{\text{Full}}$. Negative values favor the ablated variant.

The interaction study likewise shows no quadratic-schedule gain: fitness changes from 2,514.06 to 2,539.65 without local refinement and from 1,764.72 to 1,780.56 with it. Accordingly, only local refinement and elitist acceptance receive clear support; greedy evidence is mixed and OBL is independently non-significant.

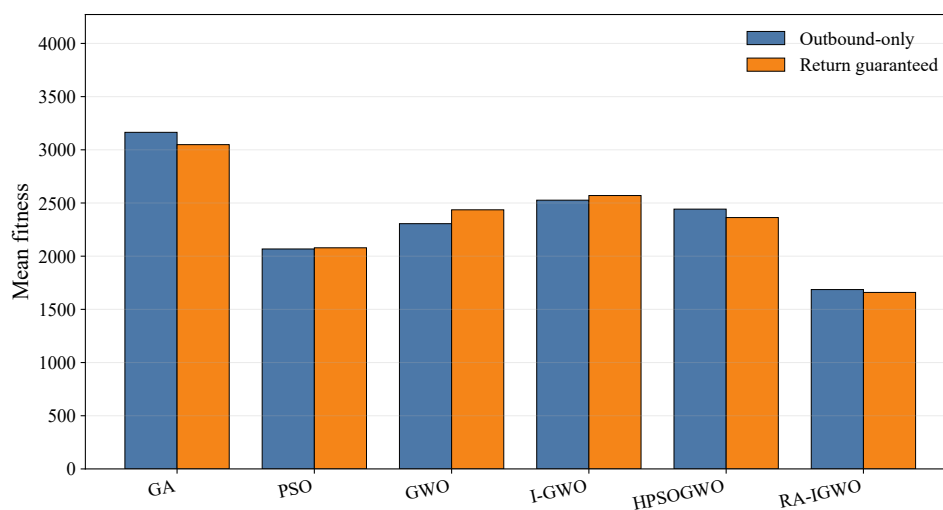
4.9 Return-to-base validation

Excluding return travel can overestimate endurance feasibility. Table 16 and Figure 4 therefore compare outbound-only and return-guaranteed evaluation for all six population methods. RA-IGWO has the lowest mean fitness and highest completion in both modes, preserving its population-baseline ranking under the more conservative endurance model.

Table 16. Outbound-only and return-guaranteed endurance evaluation

Mode	Algorithm	Fitness mean	Completion	Crossings
Outbound-only	GA	3,164.31	72.1%	1.35
Outbound-only	PSO	2,067.72	77.3%	0.25
Outbound-only	GWO	2,305.66	71.9%	0.15
Outbound-only	I-GWO	2,526.55	70.5%	0.20
Outbound-only	HPSOGWO	2,442.39	69.7%	0.25
Outbound-only	RA-IGWO	1,686.17	85.8%	0.20
Return-guaranteed	GA	3,048.62	72.1%	1.00
Return-guaranteed	PSO	2,078.84	78.1%	0.30
Return-guaranteed	GWO	2,435.93	72.3%	0.35
Return-guaranteed	I-GWO	2,570.15	68.7%	0.25
Return-guaranteed	HPSOGWO	2,363.09	70.4%	0.25
Return-guaranteed	RA-IGWO	1,659.22	85.9%	0.15

Bold indicates the lowest fitness and the highest completion rate within each endurance mode.

**Figure 4.** Effect of adding return-to-base feasibility to the endurance evaluation.

4.10 Visualization, practical relevance, and limitations

Figure 5 shows representative allocations and routes and should be read with the intersection statistics; under the soft model, crossings are penalized and reported rather than forbidden. The model addresses offline low-altitude allocation with mixed obstacles, communication degradation, and limited resources^[18,19], focusing on risk-aware ownership, route-prefix feasibility, and unified-NFE evaluation rather than communication-centric power allocation.

Future work will consider dynamically changing maps, online replanning, adaptive learning of objective weights, and comprehensive automatic tuning under equal budgets.

5. CONCLUSIONS

This paper proposed RA-IGWO for risk-aware heterogeneous MUTA under coupled spatial, communication, payload, endurance, and time-window constraints. The framework combines partial-completion evaluation with decoded-space refinement and a unified fitness-evaluation budget. Multi-instance results show robust

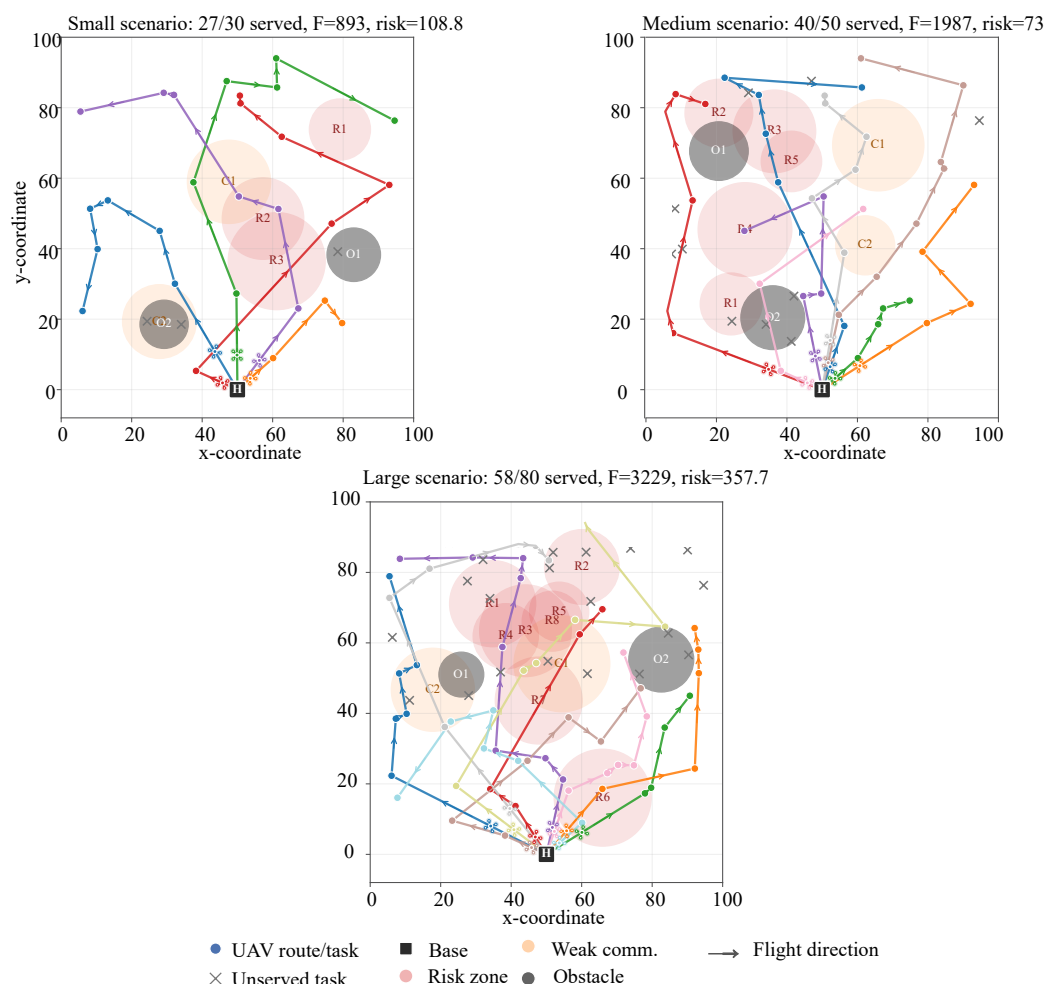


Figure 5. RA-IGWO task allocation and routes for the three scenarios. The black box marked “H” denotes the common home base (takeoff and landing pad).

advantages over population-based baselines across decoder choices, obstacle treatments, and objective-weight settings, while comparisons with D-ALNS reveal scale- and constraint-dependent performance rather than universal superiority. Ablation results identify decoded-space local refinement and accept-only-improved updates as the main supported contributors; greedy initialization and OBL show weak or mixed independent effects, whereas the quadratic schedule is not supported as a performance-enhancing component. Future work will address adaptive scheduling, systematic parameter tuning, dynamic replanning, and adaptive multi-criteria weighting.

DECLARATIONS

Authors' contributions

Conceptualization, software, methodology, writing-original draft: Miao, X.

Validation, formal analysis, review and editing: Zhang, Z.

Data curation and visualization: Wang, Y.

Project administration, funding acquisition, review and editing: Li, B.

Availability of data and materials

The raw data supporting the conclusions of this article will be made available by the authors upon request.

AI and AI-assisted tools statement

Not applicable.

Financial support and sponsorship

This work was supported by the National Natural Science Foundation of China (62473249), Natural Science Foundation of Shanghai (23ZR1426600).

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

© The Author(s) 2026.

REFERENCES

1. Chung, S.; Paranjape, A. A.; Dames, P.; Shen, S.; Kumar, V. A survey on aerial swarm robotics. *IEEE Trans. Robot.* **2018**, *34*, 837–55. [DOI](#)
2. Cao, Y.; Long, T.; Sun, J.; Wang, Z.; Xu, G. Comparison of distributed task allocation algorithms considering non-ideal communication factors for multi-UAV collaborative visit missions. *IEEE Robot. Autom. Lett.* **2025**, *10*, 1928–35. [DOI](#)
3. Gerkey, B. P.; Mataric, M. J. A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. J. Robot. Res.* **2004**, *23*, 939–54. [DOI](#)
4. Chakraborty, H.; Guérin, F.; Leclercq, E.; Lefebvre, D. Optimization techniques for multi-robot task allocation problems: review on the state-of-the-art. *Robot Autom. Syst.* **2023**, *168*, 104492. [DOI](#)
5. Choi, H. L.; Brunet, L.; How, J. P. Consensus-based decentralized auctions for robust task allocation. *IEEE Trans. Robot.* **2009**, *25*, 912–26. [DOI](#)
6. Chen, J.; Qing, X.; Ye, F.; Xiao, K.; You, K.; Sun, Q. Consensus-based bundle algorithm with local replanning for heterogeneous multi-UAV system in the time-sensitive and dynamic environment. *J. Supercomput.* **2022**, *78*, 1712–40. [DOI](#)
7. Ye, F.; Chen, J.; Sun, Q.; Tian, Y.; Jiang, T. Decentralized task allocation for heterogeneous multi-UAV system with task coupling constraints. *J. Supercomput.* **2021**, *77*, 111–32. [DOI](#)
8. Tang, J.; Chen, X.; Zhu, X.; Zhu, F. Dynamic reallocation model of multiple unmanned aerial vehicle tasks in emergent adjustment scenarios. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 1139–55. [DOI](#)
9. Holland, J. H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. Cambridge, MA: MIT Press; 1992. [DOI](#)
10. Tang, J.; Liu, D.; Wang, Q.; Li, J.; Sun, J. Probabilistic chain-enhanced parallel genetic algorithm for UAV reconnaissance task assignment. *Drones* **2024**, *8*, 213. [DOI](#)
11. Phung, M. D.; Ha, Q. P. Motion-encoded particle swarm optimization for moving target search using UAVs. *Appl. Soft Comput.* **2020**, *97*, 106705. [DOI](#)
12. Das, S.; Suganthan, P. N. Differential evolution: a survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **2011**, *15*, 4–31. [DOI](#)
13. Dorigo, M.; Maniezzo, V.; Colomi, A. Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. B* **1996**, *26*, 29–41. [DOI](#)
14. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–71. [DOI](#)
15. Chen, L.; Liu, W.; Zhong, J. An efficient multi-objective ant colony optimization for task allocation of heterogeneous unmanned aerial vehicles. *J. Comput. Sci.* **2022**, *58*, 101545. [DOI](#)
16. Han, Z.; Chen, M.; Shao, S.; Zhu, H.; Wu, Q. Cooperative multi-task assignment of unmanned autonomous helicopters based on hybrid enhanced learning ABC algorithm. *IEEE Trans. Intell. Veh.* **2024**, *9*, 526–40. [DOI](#)

17. Wolpert, D. H.; Macready, W. G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67-82. DOI
18. He, Y.; Huang, F.; Liang, Y., et al. Sum secrecy rate enhancement in low-altitude intelligent networks with mixed obstacles. *IEEE Internet Things J.* **2026**, *13*, 29969-81. DOI
19. Qin, Y.; Wang, Z.; Zhang, H. Joint task allocation and power optimization in multi-UAV Systems: an overlapping coalition formation game approach. *IEEE Trans. Veh. Technol.* **2026**, *75*, 1480-91. DOI
20. Yin, S.; Xiang, Z. Adaptive operator selection with dueling deep Q-network for evolutionary multi-objective optimization. *Neurocomputing* **2024**, *581*, 127491. DOI
21. Mirjalili, S.; Mirjalili, S. M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46-61. DOI
22. Faris, H.; Aljarah, I.; Al-Betar, M. A.; Mirjalili, S. Grey wolf optimizer: a review of recent variants and applications. *Neural Comput. Applic.* **2018**, *30*, 413-35. DOI
23. Tang, H.; Sun, W.; Lin, A.; Xue, M.; Zhang, X. A GWO-based multi-robot cooperation method for target searching in unknown environments. *Expert Syst. Appl.* **2021**, *186*, 115795. DOI
24. Nadimi-Shahraki, M. H.; Taghian, S.; Mirjalili, S. An improved grey wolf optimizer for solving engineering problems. *Expert Syst. Appl.* **2021**, *166*, 113917. DOI
25. Hou, Y.; Gao, H.; Wang, Z.; Du, C. Improved Grey wolf optimization algorithm and application. *Sensors* **2022**, *22*, 3810. DOI
26. Qiu, Y.; Yang, X.; Chen, S. An improved gray wolf optimization algorithm solving to functional optimization and engineering design problems. *Sci. Rep.* **2024**, *14*, 14190. DOI
27. Heidari, A. A.; Pahlavani, P. An efficient modified grey wolf optimizer with lévy flight for optimization tasks. *Appl. Soft Comput.* **2017**, *60*, 115-34. DOI
28. Saxena, A.; Kumar, R.; Das, S. β -chaotic map enabled grey wolf optimizer. *Appl. Soft Comput.* **2019**, *75*, 84-105. DOI
29. Tizhoosh, H. R. Opposition-based learning: a new scheme for machine intelligence. In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce. IEEE; 2005. pp. 695-701. DOI
30. Dhargupta, S.; Ghosh, M.; Mirjalili, S.; Sarkar, R. Selective opposition based grey wolf optimization. *Expert Syst. Appl.* **2020**, *151*, 113389. DOI
31. Singh, N.; Singh, S. B. Hybrid Algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. *J. Appl. Math.* **2017**, *2017*, 2030489. DOI
32. Zhang, X.; Lin, Q.; Mao, W.; Liu, S.; Dou, Z.; Liu, G. Hybrid particle swarm and grey wolf optimizer and its application to clustering optimization. *Appl. Soft Comput.* **2021**, *101*, 107061. DOI
33. Qu, C.; Gai, W.; Zhang, J.; Zhong, M. A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning. *Knowl. Based Syst.* **2020**, *194*, 105530. DOI
34. Martin, J. A.; Ouwkerk, J. N.; Lamping, A. P.; Cohen, K. Comparison of battery modeling regression methods for application to unmanned aerial vehicles. *Complex Eng. Syst.* **2022**, *2*, 5. DOI
35. Liu, J.; Wang, Y.; Xin, B.; Wang, L. A biobjective perspective for mixed-integer programming. *IEEE Trans. Syst. Man. Cybern. Syst.* **2022**, *52*, 2374-85. DOI
36. Fu, Y.; Ding, M.; Zhou, C. Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV. *IEEE Trans. Syst. Man Cybern. A* **2012**, *42*, 511-26. DOI
37. Zhang, Q.; Li, H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712-31. DOI
38. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of ICNN'95-International Conference on Neural Networks. IEEE; 1995. pp. 1942-48. DOI
39. Clerc, M.; Kennedy, J. The Particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58-73. DOI
40. Xiao, Y.; Li, Y.; Liu, H.; Chen, Y.; Wang, Y.; Wu, G. Adaptive large neighborhood search algorithm with reinforcement search strategy for solving extended cooperative multi task assignment problem of UAVs. *Inf. Sci.* **2024**, *679*, 121068. DOI

Disclaimer/Publisher's Note: All statements, opinions, and data contained in this publication are solely those of the individual author(s) and contributor(s) and do not necessarily reflect those of OAE and/or the editor(s). OAE and/or the editor(s) disclaim any responsibility for harm to persons or property resulting from the use of any ideas, methods, instructions, or products mentioned in the content.



© The Author(s) 2026. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, sharing, adaptation, distribution and reproduction in any medium or format, for any purpose, even commercially, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.