

Multiobjective multi-UAV path planning via evolutionary multitasking optimization with adaptive operator selection and knowledge fusion

Kai Meng, Binghong Wu, Bin Xin, Fang Deng, Chen Chen *

School of Automation, Beijing Institute of Technology, Beijing, 100081, China

National Key Lab of Autonomous Intelligent Unmanned Systems, Beijing, 100081, China

ARTICLE INFO

Keywords:

Multiobjective multi-UAV path planning
Evolutionary multitasking
Adaptive operator selection
Constrained multiobjective optimization

ABSTRACT

Path planning is crucial for UAV task execution, underpinning effective aerial reconnaissance and precision strikes. An ideal flight path must both minimize travel distance and reduce the risk of enemy detection or destruction. Due to the inherent trade-off between these objectives, multi-UAV path planning is conventionally formulated as a multiobjective optimization problem. However, as the number of obstacles, threats, and UAVs increases, the computational complexity escalates, hindering the generation of optimal path planning solutions via conventional multiobjective optimization approaches. To address this challenge, we model a multiobjective multi-UAV path planning (MOMUPP) problem that simultaneously optimizes flight distance and threat cost, with the latter quantified using line-of-sight theory and terrain occlusion effects. We further construct an auxiliary task that approximates the MOMUPP problem and develop an evolutionary multitasking framework to facilitate effective knowledge transfer between tasks. Building on this framework, we propose the evolutionary multitasking multiobjective path planning (EMMOP) algorithm. EMMOP incorporates a double deep Q-networks-based adaptive operator selection (DAOS) mechanism that dynamically selects the optimal search operators for each task based on the current evolutionary state, thereby generating high-quality offspring. Additionally, a knowledge transfer strategy based on directional information extraction and knowledge fusion (KTDF) enables efficient exchange of critical information between the main and auxiliary tasks. Experiments on 15 benchmark instances across five map scenarios indicate that EMMOP outperforms five state-of-the-art methods, enhancing hypervolume by 2.46% and pure diversity by 28.27%, while generating shorter, safer, and collision-free UAV paths with diverse trade-off solutions for decision-makers. The source code is available at <https://github.com/Leopard125/EMMOP>.

1. Introduction

In the last decade, unmanned aerial vehicles (UAVs) have become increasingly prevalent in both civilian and military domains due to their compact size and high maneuverability [1–3]. As mission requirements grow more complex, multi-UAV systems have demonstrated significant advantages over single-UAV operations, particularly for large-scale and high-risk tasks. In such scenarios, efficient and reliable path planning becomes critical to ensure coordinated behavior and mission success [4,5].

Path planning seeks to determine a safe and collision-free path from origin to destination [6]. For UAVs, this process must account for various factors, including path length, flight altitude, avoidance of restricted areas, and the risk of terrain collisions. Compared to

single-UAV scenarios, multi-UAV path planning introduces additional challenges, particularly the need to avoid inter-vehicle collisions during cooperative operations. Addressing these issues requires not only the identification of non-dominated solutions across multiple conflicting objectives, but also strict adherence to UAV maneuverability constraints and environmental safety requirements. Given the increased dimensionality and computational demands imposed by these factors, the development of robust and scalable path planning algorithms becomes essential.

Current methods for multi-UAV path planning include graph-based searches [7], sampling techniques [8], potential field approaches [9], and meta-heuristics [10]. Most of these approaches adopt a single-objective formulation or rely on weighted-sum methods to handle

* Corresponding author at: School of Automation, Beijing Institute of Technology, Beijing, 100081, China.

E-mail addresses: 3120195451@bit.edu.cn (K. Meng), 3120240820@bit.edu.cn (B. Wu), brucebin@bit.edu.cn (B. Xin), dengfang@bit.edu.cn (F. Deng), xiaofan@bit.edu.cn (C. Chen).

multiple objectives. Although such techniques are computationally efficient, they require predefined weight vectors, which are often difficult to specify in complex mission scenarios due to uncertain decision-maker preferences. In contrast, modeling the problem as a constrained multi-objective optimization problem (CMOP) offers a more expressive framework. It enables the generation of diverse Pareto-optimal solutions that better capture the inherent trade-offs among conflicting objectives and provide greater flexibility for informed decision-making.

Despite its practical importance, research on multi-objective multi-UAV path planning (MOMUPP) remains relatively limited. Early studies, such as Zhang et al. [11], adopted MOEA/D-based approaches to formulate MOMUPP as a CMOP, aiming to balance competing objectives. However, many existing models [11–15] simplify threat cost estimation by considering only the distance to threat sources, while neglecting critical environmental factors such as terrain occlusion. This simplification reduces the accuracy of threat representation. In realistic scenarios, terrain features can obstruct the line-of-sight (LoS) between UAVs and ground-based radars, thereby diminishing actual threat exposure. By incorporating LoS-based terrain occlusion into threat modeling, it becomes possible to identify terrain-shielded regions, leading to safer path planning and improved solution quality.

However, this enhanced modeling, while improving path safety, introduces notable challenges for optimization. To avoid threats, some paths may be fully confined within occluded regions. Although this strategy effectively minimizes threat-related costs, it often results in excessively long paths, thereby stretching one end of the Pareto front (PF) and forming a long tail structure that compromises the uniformity of solution distribution. Conversely, when paths attempt to traverse occluded areas to exploit terrain occlusion, the complex structure of occlusions fragments the otherwise continuous feasible space into multiple disjoint regions. This fragmentation increases the risk of the algorithm becoming trapped in local optima and significantly complicates the maintenance of both diversity and convergence during the optimization process.

To address these issues, constrained multiobjective evolutionary algorithms (CMOEAs) [16], such as NSGA-II [14] and various multi-objective swarm intelligence algorithms [17,18], have been employed for MOMUPP. Although effective in certain settings, these algorithms typically adopt a single-population model with multiple local search operators, limiting their global search ability and making them prone to premature convergence—especially when handling irregular PFs induced by terrain occlusion.

Evolutionary multitasking (EMT) [19,20] has recently emerged as a promising paradigm for improving search efficiency in complex optimization problems. By introducing multiple related auxiliary tasks, EMT reformulates challenging optimization problems into multitasking problems, thereby facilitating knowledge transfer and enhancing both efficiency and solution quality. Rather than addressing problems in isolation, EMT exploits shared information to minimize redundant exploration and reduce computational costs. Nevertheless, despite the demonstrated competitiveness of existing EMT algorithms in practical engineering applications [21], several critical challenges remain unresolved. These include the design of auxiliary tasks that closely reflect the main task's characteristics, the efficient extraction and utilization of valuable information from these tasks, and the selection of suitable evolutionary operators that maintain optimization stability.

Based on these considerations, this article proposes an evolutionary multitasking multiobjective path planning (EMMOP) algorithm tailored to solve MOMUPP problems. The key contributions are summarized as follows:

(1) A variant of the MOMUPP problem is investigated, where terrain occlusion is incorporated to improve the realism of threat modeling. A threat cost model is constructed based on LoS theory and terrain occlusion, enhancing flight safety while enabling infeasible regions to be identified and redundant exploration to be reduced during optimization.

(2) A simple auxiliary task related to the MOMUPP problem is introduced within a novel EMT framework. This framework reformulates MOMUPP as a multitasking optimization problem, thereby facilitating effective knowledge transfer from the auxiliary task to the main task and enhancing computational efficiency and scalability.

(3) An EMMOP algorithm is proposed that integrates the DAOS and KTDF strategies to efficiently solve MOMUPP problems within EMT framework. DAOS adaptively selects the optimal operator for each task to generate high-quality offspring, while KTDF employs an adaptive fusion matrix to capture and integrate search direction information, ensuring effective knowledge transfer.

The remainder of this study is structured as follows: Section 2 reviews the related work on multi-UAV path planning methods, as well as the methods of EMT and adaptive operator selection. Section 3 outlines the mathematical model for MOMUPP. The details of the proposed EMMOP algorithm are presented in Section 4. Section 5 discusses the simulation results and their analysis, while Section 6 concludes with a summary and suggestions for future work.

2. Related works

2.1. Multi-UAV path planning methods

Multi-UAV path planning poses a significant challenge in autonomous systems, requiring a trade-off between multiple conflicting objectives such as path length, energy efficiency, and collision avoidance under complex constraints [22]. Existing methods are generally divided into two main categories: weighted multiobjective (WMO) algorithms and Pareto-based multiobjective (PMO) algorithms.

2.1.1. WMO algorithms

The WMO approach addresses conflicting objectives by assigning weights to reflect their relative importance, thereby transforming them into a single objective using a weighted sum. For instance, Zhang et al. [23] introduced an interfered fluid dynamical system (IFDS) for UAV formation path planning, where the weighted sum of path length, collision avoidance, path smoothness, required overload, and tracking error is evaluated. Despite its effectiveness, this approach can lead to significant computational expenses, especially under increasingly complex constraints and environmental conditions.

Meta-heuristic algorithms, including particle swarm optimization (PSO) [24–27], artificial ecosystem optimizer [28], and cuckoo search [29] have been widely adopted in multi-UAV path planning. Compared to traditional methods, these meta-heuristics are capable of generating approximate solutions with lower computational cost. For example, Liu et al. [24] developed an enhanced PSO algorithm that employs a spatially refined voting mechanism to address six weighted objectives: minimum threat, height, fuel supply, time coordination, collision avoidance, and UAV maneuverability. Shao et al. [25] proposed a comprehensively enhanced PSO to optimize the weighted sum of path length, terrain, radar detection, and collision costs. He et al. [26] combined PSO with an improved symbiotic organisms search algorithm to address multi-UAV cooperative path planning, formulating the fitness function as a weighted sum of path length, altitude, smoothness, and threat level. Xu et al. [27] further improved PSO by incorporating a dynamic multi-swarm mechanism aimed at optimizing path length, turns, collisions, separation maintenance, and communication maintenance. Additionally, Niu et al. [28] integrated an enhanced artificial ecosystem optimizer with reinforcement learning (RL) to optimize the weighted sum of terrain, path length, threat, smoothness, and collision avoidance, facilitating collaborative paths for multiple UAVs. Lou et al. [29] presented a hybrid method that combines cuckoo search and golden jackal optimization (CS-GJO) for multi-UAV cooperative trajectory planning, optimizing the weighted sum of threat, fuel, altitude, smoothness, and temporal-spatial collaboration. Despite these advances, WMO algorithms often produce suboptimal results when dealing with conflicting objectives due to the difficulty in setting appropriate weight values.

Table 1
Analysis of multi-UAV path planning problems and employed methods

Method	Literature	Algorithm	EMT	RL-AOS	Objectives	Constraints
WMO	Zhang et al. [23]	IFDS	N	N	Path length, collision avoidance, smoothness, overload, tracking error	–
	Liu et al. [24]	PSO	N	N	Threat, altitude, fuel, time, collision avoidance, maneuverability	–
	Shao et al. [25]	CIPSO	N	N	Path length, terrain, radar, collision	–
	He et al. [26]	HIPSO-MSOS	N	N	Path length, threat, smoothness, altitude	–
	Xu et al. [27]	DMSPSO	N	N	Path length, turns, collision, separation, communication	–
	Niu et al. [28]	MEAEO-RL	N	Y	Path length, terrain, threat, smoothness, collision	–
PMO	Lou et al. [29]	CS-GJO	N	N	Threat, fuel, altitude, smoothness, time-space collaboration	–
	Zhou et al. [14]	TSKAC-NSGA-II	N	Y	Energy, flight safety	Collision, threat, maneuverability
	Zhang et al. [12]	MCMOPSO-RL	N	Y	Distance, constraint satisfaction	–
	Bai et al. [13]	DSNSGA-III	N	N	Path length, time, threat, cooperation	Maneuverability
	Zhong et al. [15]	INSGA-II	N	N	Path length, altitude, constraint violations	–
	Zhang et al. [11]	Improved MOEA/D	N	N	Path length, terrain avoidance	Collision, altitude, maneuverability
This Paper		EMMOP	Y	Y	Path length, threat cost (terrain occlusion)	Terrain, collision, maneuverability

2.1.2. PMO algorithms

Unlike WMO algorithms, PMO algorithms aim to generate a diverse set of Pareto-optimal solutions, offering more flexibility for decision-makers to explore trade-offs among conflicting objectives. Various optimization methods, such as NSGA-II, MOEA/D, and their variants, have been proposed. Most existing research has concentrated on developing search mechanisms and constraint handling techniques (CHTs). For example, Zhang et al. [12] developed the multiobjective PSO, MCMOPSO-RL, which integrates reinforcement learning to simultaneously optimize flight and constraint costs. Bai et al. [13] applied a many-objective evolutionary algorithm to concurrently optimize multiple objectives, such as path length, flight time, threat level, and cooperative distance. Zhou et al. [14] employed a two-stage, knowledge-assisted coevolutionary NSGA-II to optimize both UAV energy consumption and safety objectives. Zhong et al. [15] introduced a novel NSGA-II (INSGA-II) featuring an advanced crowding distance mechanism for path planning of multiple UAVs in combat scenarios. They modeled the mission as a multiobjective optimization problem aimed at minimizing length, altitude, and constraint violations. Zhang et al. [11] presented a MOEA/D method incorporating a dual-CHT strategy, aiming to minimize flight distance while ensuring terrain avoidance. Despite their effectiveness in generating well-balanced solutions, PMO algorithms often suffer from significantly higher computational costs, as they typically initiate the optimization process from scratch for each problem instance. Furthermore, most existing approaches do not leverage transferable knowledge across different MOMUPP scenarios, thereby limiting their search efficiency and adaptability in a more complex solution space.

Table 1 summarizes recent multi-UAV path planning studies and highlights a key observation: most existing approaches simplify threat modeling by estimating threat costs solely based on distance to threat sources, neglecting the critical impact of terrain occlusion. In real-world missions, threats are often partially obscured by terrain features, and failing to account for this can result in overly conservative or unsafe paths. Although integrating terrain occlusion into threat modeling improves planning realism and mission success rates, it also introduces significant challenges: increased computational complexity, less regular PFs, and reduced optimization efficiency—particularly for both WMO and PMO algorithms. These issues are further exacerbated in the MOMUPP with terrain occlusion, where multiple objectives and constraints interact nonlinearly. Given these limitations, there is a pressing need to construct a more accurate threat cost model that explicitly accounts for terrain occlusion effects. Furthermore, it is essential to design an EMT algorithm capable of effectively handling the added complexity of the MOMUPP problem under such constraints.

2.2. Evolutionary multitasking

In real-world applications, problems are often interconnected, with insights from one problem aiding the resolution of others. EMT optimization has garnered significant attention due to its capacity to tackle multiple problems simultaneously, leveraging knowledge transfer between tasks to enhance performance. This paper introduces key EMT algorithms, which can generally be categorized into two types: single-population methods and multi-population methods.

The multifactorial evolutionary algorithm (MFEA) [30], proposed by Gupta et al. was one of the earliest EMT methods designed to optimize multiple problems using a single population. In MFEA, individuals are represented within a unified search space and subsequently decoded into task-specific solutions. Knowledge transfer occurs implicitly across tasks through adaptive crossover and mutation mechanisms. Building on MFEA, Gupta et al. [31] presented a multiobjective multi-factorial evolutionary algorithm (MO-MFEA), marking the first application of EMT to multiobjective optimization. Similarly, Yang et al. [32] integrated a surrogate model into MFEA to enhance inter-task knowledge transfer by reinforcing the connections between tasks.

In contrast to single-population EMT methods, which rely on genetic operators to facilitate implicit knowledge transfer, multi-population EMT approaches maintain distinct populations for each task, enabling explicit optimization tailored to each task. The efficiency of knowledge transfer, a core element of EMT, has been extensively studied, with research aimed at enhancing positive transfer while minimizing detrimental effects. For instance, Wu et al. [33] introduced an evolutionary state estimator mechanism to improve transferability between tasks with distinct characteristics. Li et al. [34] proposed diversity-oriented and convergence-oriented knowledge transfer strategies to optimize the identification of high-potential solutions for transfer. Additionally, given the diverse characteristics of task domains, not all source tasks provide equal benefits for the optimization of target tasks. As a result, numerous studies have investigated auxiliary task selection strategies, including those based on task relationships [35], task decomposition [36], adaptive task selection [37], and historical experiences [38].

EMT methods have been utilized to address a range of benchmark challenges, such as constrained multiobjective optimization [20], interval multiobjective optimization [39], and expensive minimax optimization [40]. In practical engineering environments, such as sensor deployment [41], feature selection [42], edge computing [43], vehicle routing problem [44], and beneficiation processes [45], EMT has also demonstrated significant effectiveness.

Despite the success of EMT, its application to MOMUPP remains largely underexplored. This problem presents significant challenges for

EMT, particularly in handling complex constraints and extracting useful knowledge from infeasible regions. Moreover, existing methods are ill-suited for high-dimensional MOMUPP scenarios, as they typically employ the same optimizer for both auxiliary and main tasks. This approach limits search diversity and offers no guarantee of improving the feasibility or quality of solutions. To overcome these limitations, we propose a robust EMT-based framework that integrates DRLOS and KTDF, aiming to enhance both solution quality and computational efficiency for MOMUPP.

2.3. RL-based adaptive operator selection

RL-based adaptive operator selection (AOS) [46,47] is an advanced framework that integrates reinforcement learning into the operator selection process within evolutionary algorithms and metaheuristics. Its primary objective is to dynamically identify the most effective search operator during optimization. In this framework, operator selection is typically formulated as a Markov Decision Process (MDP), where each operator corresponds to an action taken at a given state in the search process. As operators generate new solutions, the search state evolves, and a reward function evaluates each operator's effectiveness. Compared to traditional AOS, RL-based AOS leverages informative state features and a well-trained model to rapidly adapt to operator performance variations, thereby improving convergence and solution quality.

Recent studies have extensively explored RL-based AOS, with early research employing tabular RL methods such as SARSA [48] and Q-learning [49] in evolutionary algorithm design. However, these approaches suffer from scalability limitations due to their reliance on state-action tables, which become impractical as the state and action spaces grow. With advances in computational power, deep reinforcement learning (DRL) has emerged as a viable solution, utilizing neural networks for state-action mapping to enhance scalability and adaptability in complex environments. Notably, efficient DRL-based methods such as deep Q-networks (DQN) [49], dueling deep Q-network [50], and double deep Q-networks (D2QN) [51] have been successfully applied to AOS.

Building on these advanced RL techniques, RL-based AOS has demonstrated versatility and effectiveness across various application domains. In path planning, Qu et al. [52] employed Q-learning to optimize grey wolf optimizer operations, thereby enhancing path quality. Yu and Luo [53] introduced an RL-based method for dynamically adjusting search strategies and switching parameters in the cuckoo search algorithm, significantly improving convergence speed. This approach has been successfully utilized in UAV path planning, ensuring the generation of reliable and secure flight trajectories. In UAV formation path planning, Wang et al. [54] developed an RL-based artificial rabbit optimizer that adaptively selects optimal strategies at different evolutionary stages, significantly enhancing algorithm performance in complex environments such as mountainous forests. Zhan et al. [55] integrated Q-learning with genetic algorithms to enhance search and rescue path planning, optimizing offspring evolution across generations. Beyond path optimization, RL-driven AOS has been effectively employed in various domains, such as job-shop scheduling [56], satellite scheduling [57], vehicle routing [58], and railway catenary deicing [59].

These RL-based AOS approaches offer a promising means for designing differentiated optimizers for the main and auxiliary tasks. However, several challenges remain. One key limitation is the limited diversity of the operator pool, which constrains the representation of the state space. Moreover, as optimization progresses, the reward function tends to become sparse, reducing the agent's ability to explore effectively. To overcome these issues, this work enhances the action, state, and reward mechanisms. Specifically, a D2QN is employed to adaptively select the optimal search operator for each task. Given the continuous state space and discrete action space in MOMUPP, D2QN is particularly well-suited to this problem.

2.4. Motivation

Existing studies on multi-UAV path planning have primarily focused on threat avoidance by maximizing the distance from hostile zones. While this strategy is effective to a certain extent, it often overlooks the crucial impact of terrain occlusion—an essential factor in real-world threat environments. Neglecting terrain occlusion can result in oversimplified threat models that fail to capture the complexity and realism of operational scenarios.

To handle multi-objective path planning problems, various WMO and PMO algorithms have been developed to balance convergence, diversity, and constraint feasibility, with some also leveraging infeasible solutions to guide the search process. However, these algorithms are typically designed to address a single task per run and lack the capability for cross-task knowledge reuse. This limitation becomes particularly pronounced when tackling complex problems such as MOMUPP with terrain occlusion, where adaptability and knowledge generalization across related instances are essential.

In response to this limitation, EMT has emerged as a promising paradigm, offering the potential to solve multiple related tasks simultaneously through knowledge transfer. However, most existing EMT approaches concentrate on auxiliary task selection and knowledge transfer design, while overlooking the fact that the main and auxiliary tasks may benefit from distinct optimization dynamics. Applying a unified optimization strategy across tasks can restrict diversity and hinder effective exploration of complementary search spaces.

To address these challenges, this study proposes a novel threat cost model that explicitly incorporates terrain occlusion effects, thereby enhancing the realism and safety of multi-UAV path planning in complex environments. In addition, an evolutionary multitasking multi-objective path planning algorithm (EMMOP), specifically designed for the MOMUPP problem, is introduced. EMMOP integrates a DAOS strategy with a KTDF mechanism. This integration enables effective cross-task knowledge transfer, which promotes the efficient exploration of promising regions and supports the generation of high-quality, non-dominated solutions with reduced computational overhead.

3. Problem description

3.1. Path representation

In this study, we utilize a spherical coordinate system to define the UAV's spatial positions. A complete path W_i from the starting point S to the target point T is expressed as:

$$W_i = \{S, (\theta_{i,1}, \omega_{i,1}, \varphi_{i,1}), \dots, (\theta_{i,D}, \omega_{i,D}, \varphi_{i,D}), T\}, \quad (1)$$

where θ , ω , and φ indicate magnitude, elevation angle and azimuth angle, respectively. These spherical coordinates are transformed into Cartesian coordinates (x, y, z) using the following transformation matrix:

$$\begin{bmatrix} \theta \\ \omega \\ \varphi \end{bmatrix} = \begin{bmatrix} \sin \omega \cos \varphi & \sin \omega \sin \varphi & \cos \omega \\ \cos \omega \cos \varphi & \cos \omega \sin \varphi & -\sin \omega \\ -\sin \varphi & \cos \varphi & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (2)$$

Here, $i = 1, 2, \dots, K$, $j = 1, 2, \dots, D$, with (x, y, z) denoting the Cartesian coordinates of the waypoints.

3.2. MOMUPP model

3.2.1. Objective function

This study develops a constrained multiobjective optimization model that simultaneously optimizes both flight length and threat cost. The path length cost, denoted as f_1 , is defined as follows:

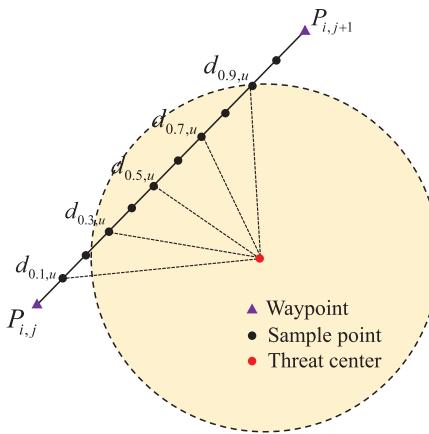


Fig. 1. Schematic diagram of the threat cost.

$$f_1(W_i) = \sum_{j=1}^{D+1} \left\| \overrightarrow{P_{i,j} P_{i,j+1}} \right\|, \quad (3)$$

where $P_{i,j}$ represents the j th waypoint of the i th UAV's path.

The threat cost associated with a UAV is estimated using an approximation method that evaluates each line segment between consecutive waypoints individually. As illustrated in Fig. 1, ten equally spaced sample points are generated along each segment connecting points i and j , from which five representative points are selected to approximate the threat level [60]. The traditional distance-based threat cost is defined as:

$$f_2(W_i) = \sum_{j=1}^D Th(j, j+1),$$

$$Th(j, j+1) = \begin{cases} \frac{\left\| \overrightarrow{P_{i,j} P_{i,j+1}} \right\|}{5} \cdot \sum_{u=1}^{N_{\text{threat}}} le_u \cdot \left(\frac{1}{d_{0.1,u}^4} + \frac{1}{d_{0.3,u}^4} + \frac{1}{d_{0.5,u}^4} + \frac{1}{d_{0.7,u}^4} + \frac{1}{d_{0.9,u}^4} \right), & \text{if } D_{u-\text{uav}} \leq r_{\det}^{\max}, \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where N_{threat} denotes the number of threat sources, and $d_{q,u}$ is the distance from the u th threat center to the sample point at relative position $q \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ along the segment $\left\| \overrightarrow{P_{i,j} P_{i,j+1}} \right\|$. The term $D_{u-\text{uav}}$ indicates minimum distance between the path segment and the u th threat source, le_u is the threat intensity of source u , and r_{\det}^{\max} is the maximum effective detection radius.

However, this model does not account for the impact of terrain on threat effectiveness. In realistic environments, terrain features can block the LoS between the threat and the UAV, forming blind zones that significantly reduce the risk of detection. To address this, a radar detection probability model incorporating terrain occlusion is introduced, defined as:

$$R_u^{\text{rad}} = \begin{cases} 0, & \text{if } (D_{u-\text{uav}} > r_{\det}^{\max}) \text{ or } (\text{LoS} = 0) \\ \frac{1}{1 + (p_1 d_u^4 / \text{RCS})^{p_0}}, & \text{otherwise,} \end{cases} \quad (5)$$

where p_0 and p_1 are radar model parameters, d_u denotes the distance between the UAV and the u th radar center, and RCS represents the radar cross-section of the UAV, which is computed following the method in [61]. The LoS condition is defined as:

LoS =

$$\begin{cases} 1, & \text{if } \forall (x_c, y_c, z_c) \in C \text{ and } \text{dem}(x_c, y_c) \leq z_c \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Here, $\text{dem}(\cdot)$ denotes the digital elevation model function, C represents the collection of LoS points between the UAV and the radar, and

(x_c, y_c, z_c) corresponding to the coordinates of a LoS point. Specifically, LoS equals 0 if terrain obstructs the path between the UAV and the radar; otherwise, LoS equals 1. A visual representation of this mechanism is provided in Fig. 2.

By incorporating terrain occlusion, the revised model more accurately captures the UAV's exposure to radar detection and enables safer trajectory planning. The updated threat cost function is formulated as:

$$f_2(W_i) = \sum_{j=1}^D \frac{\left\| \overrightarrow{P_{i,j} P_{i,j+1}} \right\|}{5} \cdot \left(1 - \prod_{u=1}^{N_{\text{threat}}} (1 - R_u^{\text{rad}}) \right). \quad (7)$$

To rigorously compare the terrain occlusion-based threat model (Eq. (7)) and the traditional distance-based threat model (Eq. (4)), we conduct a comparative experiment in a synthetic terrain scenario with multiple threat sources. As shown in Fig. 3, two heatmaps visualize the spatial distribution of threat values computed by each model.

Fig. 3(a) presents the threat distribution derived from the traditional model. In this case, threat intensity is solely determined by the radial distance from each threat source, resulting in uniformly expanding circular regions of influence. Due to the overlap of multiple detection zones, large contiguous areas of high threat (indicated in yellow) emerge. In contrast, Fig. 3(b) displays the results of our proposed model, which incorporates both RCS effects and terrain occlusion. By leveraging a digital elevation map, the model accurately evaluates LoS conditions and significantly attenuates threat values in areas where terrain obstructs visibility. Consequently, the generated threat distribution is more realistic and spatially heterogeneous, with a substantial reduction in the size and density of high-threat zones.

3.2.2. Constraint condition

The multi-UAV path planning must satisfy the following constraints:

(1) Physical capabilities constraints

The first constraint ensures that the UAV's turning and climbing angles stay within feasible limits. Specifically, the turning angle between consecutive segments must not exceed the maximum turning angle α_{\max} , and the climbing angle must remain below the maximum climbing angle β_{\max} . These constraints are defined as follows:

$$C_1(W_i) = \max_{j \in [1, D]} (\alpha_{i,j} - \alpha_{\max}) \leq 0, \quad (8)$$

$$\alpha_{i,j} = \arctan \left(\frac{\left\| \overrightarrow{P'_{i,j-1} P'_{i,j}} \times \overrightarrow{P'_{i,j} P'_{i,j+1}} \right\|}{\left\| \overrightarrow{P'_{i,j-1} P'_{i,j}} \cdot \overrightarrow{P'_{i,j} P'_{i,j+1}} \right\|} \right), \quad (9)$$

$$C_2(W_i) = \max_{j \in [1, D]} (\beta_{i,j} - \beta_{\max}) \leq 0, \quad (10)$$

$$\beta_{i,j} = \arctan \frac{z_{i,j+1} - z_{i,j}}{\left\| \overrightarrow{P'_{i,j-1} P'_{i,j}} \right\|}, \quad (11)$$

where $\alpha_{i,j}$ and $\beta_{i,j}$ represent the turning and climbing angles at the j th waypoint in the path of the i th UAV, respectively, as shown in Fig. 4. The vectors $\overrightarrow{P'_{i,j-1} P'_{i,j}}$ and $\overrightarrow{P'_{i,j} P'_{i,j+1}}$ stand for projections of the consecutive path segments $\overrightarrow{P_{i,j-1} P_{i,j}}$ and $\overrightarrow{P_{i,j} P_{i,j+1}}$ onto the horizontal plane, respectively. $z_{i,j}$ signifies the z-coordinates of the j th waypoint in the i th UAV. (2) Collision constraint

The second constraint addresses the potential risk of collisions between UAVs. To avoid collisions, two conditions must be met: (1) the distance between the waypoints of different UAVs must exceed a safe distance d_s , and (2) the time difference between UAVs at these waypoints must be greater than the permissible time difference t_s . The collision constraint is expressed as:

$$C_3 = \sum_{i=1}^{K-1} \sum_{m=i+1}^K \sum_{j=1}^{D-1} \sum_{n=1}^{D-1} o_{i,m,j,n} \leq 0, \quad (12)$$

$$o_{i,m,j,n} = \begin{cases} d_s - d_{i,m,j,n}, & \text{if } d_{i,m,j,n} < d_s \text{ and } t_{i,m,j,n} < t_s \\ 0, & \text{otherwise,} \end{cases} \quad (13)$$

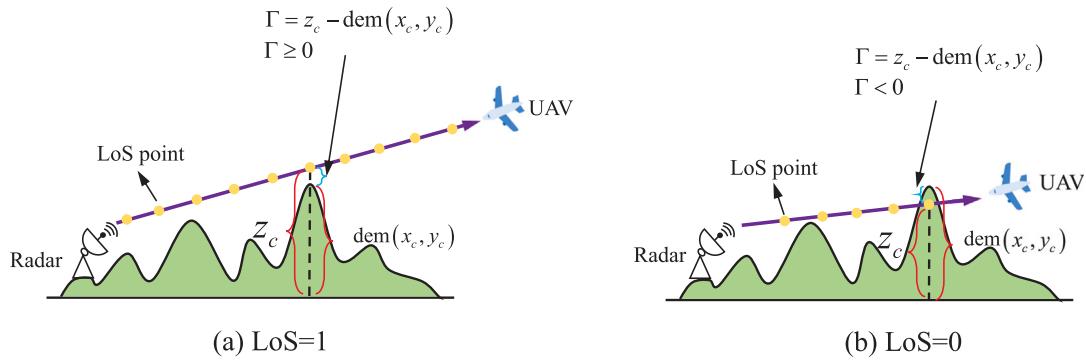


Fig. 2. Schematic diagram of calculation of the terrain occlusion mechanism.

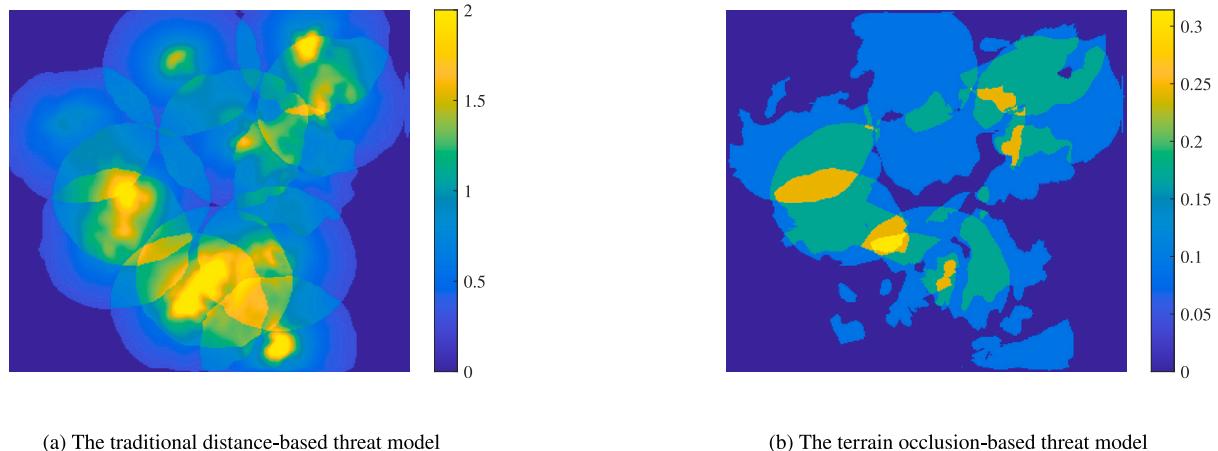


Fig. 3. Comparison of threat models via heatmaps.

$$d_{i,m,j,n} = \sqrt{(x_{i,j} - x_{m,n})^2 + (y_{i,j} - y_{m,n})^2 + (z_{i,j} - z_{m,n})^2}, \quad (14)$$

$$t_{i,m,j,n} = \left| t_{i,j} - t_{m,n} \right|, \quad (15)$$

where $t_{i,j}$ and $t_{m,n}$ are the times when i th and m th UAVs reach their respective waypoints $P_{i,j}$ and $P_{m,n}$, and $d_{i,m,j,n}$ indicates the distance between them.

(3) Terrain constraint

The third constraint ensures that the UAVs maintain a minimum altitude H_{ft} above the terrain, given by:

$$C_4(W_i) = H_{ft} - \min_{j \in [1, D]} (z_{i,j} - H_t(x_{i,j}, y_{i,j})) \leq 0, \quad (16)$$

where $H_t(x_{i,j}, y_{i,j})$ represents the terrain elevation at the coordinates $(x_{i,j}, y_{i,j})$.

In summary, the MOMUPP model is formulated as:

$$\begin{aligned} \min F(W) &= (f_1(W), f_2(W)), \\ \text{s.t. } & \begin{cases} C_q(W) \leq 0, \quad q = 1, \dots, Y, \\ W \in \Re^G, \end{cases} \end{aligned} \tag{17}$$

where W stands for a G -dimensional decision vector, encoded as shown in Fig. 5, with $G = 3 \cdot K \cdot D$ representing the search space.

3.3. Auxiliary task construction

The MOMUPP problem is inherently complex due to the fragmented feasible regions caused by stringent spatial constraints. This results in two main challenges: (i) premature convergence to local PFS due to isolated feasible subregions, and (ii) limited exploration capacity, which is a consequence of traditional feasibility-driven selection mechanisms.

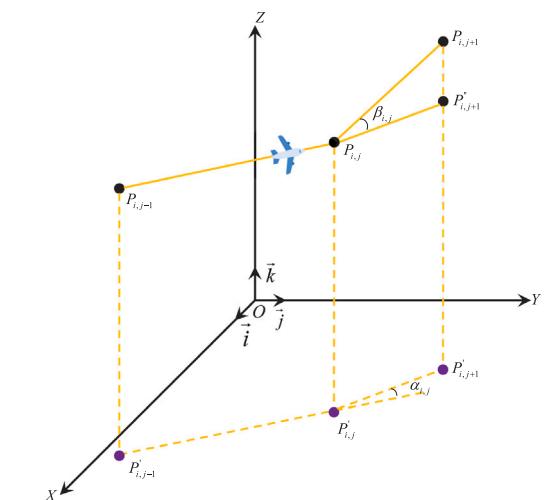


Fig. 4. Graphical illustration of the climbing and turning angles.

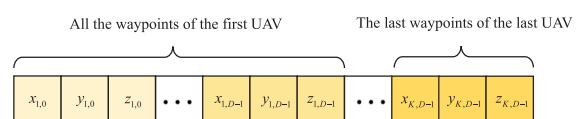


Fig. 5. Decision vector encoding diagram.

To address these challenges, we draw inspiration from EMT to enhance exploration through knowledge transfer between the main and auxiliary tasks. This approach allows the computationally demanding main

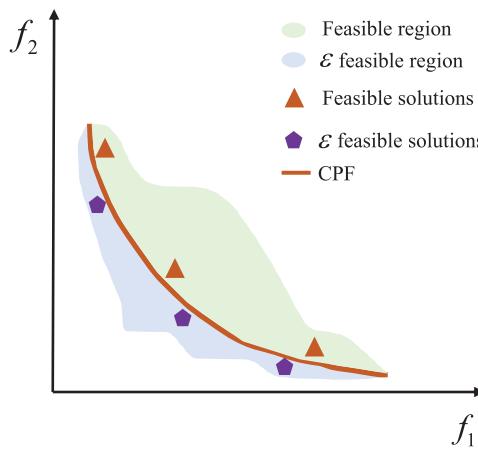


Fig. 6. The individual saved in the auxiliary task.

task to benefit from related, lower-cost tasks, ultimately improving convergence toward the global optimum. Accordingly, we construct an auxiliary task, referred to as MOMUPP-A, by relaxing the constraints of the original problem to facilitate the solution of MOMUPP, as described below:

$$\begin{aligned} \min F(W) &= (f_1(W), f_2(W)), \\ \text{s.t. } C_q(W) &\leq \varepsilon(g), \end{aligned} \quad (18)$$

where $q=1,2,3,4$, $W \in \mathbb{R}^G$, g denotes the current generation. The term $\varepsilon(g)$ represents the relaxed constraint boundary, defined as:

$$\varepsilon(g) = \begin{cases} \varepsilon(0)(1 - g/g_{max})^{\hat{r}}, & \text{if } 0 \leq g \leq g_{max}, \\ 0, & \text{if } g \geq g_{max}, \end{cases} \quad (19)$$

where \hat{r} controls the degree of constraint convergence, g_{max} represents the maximum number of iterations, and $\varepsilon(0)$ denotes the initial constraint violation, calculated as:

$$\varepsilon(0) = \max \{ \max \{ 0, C_q(W_k) \} \}, \quad (20)$$

where W_k stands for the top k th individual.

As illustrated in Fig. 6, the auxiliary task preserves solutions that are feasible within $\varepsilon(g)$, enabling interaction with feasible solutions. This process generates offspring with greater population diversity, thereby mitigating premature convergence. While the two tasks focus on different objectives—one on feasible solutions and the other on exploring infeasible regions—they collaborate synergistically, ultimately converging on the same optimal solution.

4. Proposed EMMOP

This section offers a detailed explanation of the proposed EMMOP algorithm, designed to address the MOMUPP problem. First, the overall framework of EMMOP is presented. Next, the key components are detailed, including the EMT framework, the DAOS-based offspring generation, and the knowledge transfer mechanism, which leverages directional information extraction and knowledge fusion techniques. Finally, the computational complexity of EMMOP is analyzed.

4.1. Framework of EMMOP

The overall framework of the proposed algorithm is depicted in Fig. 7 and consists of three principal components: (1) the EMT framework, (2) a double deep Q-networks-based adaptive operator selection (DAOS), and (3) a knowledge transfer strategy based on directional information extraction and knowledge fusion (KTDF). Initially, the EMT framework initializes two populations, each assigned to a specific optimization task. The DAOS mechanism is then applied to guide the generation of offspring. During the entire evolutionary process, the KTDF

strategy enables inter-task knowledge transfer by selectively extracting and fusing valuable evolutionary information from auxiliary tasks, thereby assisting the main task in exploring more promising regions of the search space and improving overall optimization performance.

Details of EMMOP algorithm are outlined in Algorithm 1. The optimization process begins with the initialization of the main task population, denoted as P_m , which is responsible for exploring and identifying the constrained PF (CPF) (Line 1). To facilitate more effective exploration, an auxiliary task, closely related to the main task, is introduced using Eq. (18) (Line 2). The corresponding auxiliary population, P_a , is then initialized to provide additional search guidance (Line 3). Both populations are then evaluated based on their respective objectives (Line 4).

Next, offspring populations Off_m and Off_a are generated according to the AOS mechanism outlined in Algorithm 2 (Line 8). The main task population P_m is combined with its offspring to form a temporary population PO_m , while the auxiliary population P_a is merged with its offspring to create a temporary population PO_a (Lines 9–10). Each member of the temporary populations undergoes assessment using the fitness evaluation method employed in SPEA2 (Line 11). Following this, knowledge transfer individuals are generated based on Algorithm 3 (Line 12).

In the final stage, the temporary populations PO_m and PO_a , along with the transferred individuals, are merged into the updated populations P'_m and P'_a (Lines 13–14). These populations undergo environmental selection to determine the parent population for the next generation. Specifically, the main task uses a selection mechanism based on the CDP method to ensure a well-distributed and high-quality non-dominated solutions (Line 15). Meanwhile, the auxiliary task employs an enhanced ε -constraint approach to refine feasible solutions (Line 16). These steps are repeated until the predefined maximum number of function evaluations is reached.

Algorithm 1 Pseudocode of EMMOP

```

Input: Population size ( $L$ ), Maximum function evaluations ( $MaxEval$ ).
Output: The non-dominated solutions from  $P_m$ .
1: Initialize the main task population  $P_m$  with size  $L$ ;
2: Generate auxiliary tasks based on Eq. (18);
3: Initialize the auxiliary task population  $P_a$  with size  $L$ ;
4: Evaluate all individuals in  $P_m$  and  $P_a$ ;
5: Set  $\varepsilon(0) = \max \{ \max \{ 0, C_q(W_k) \} \}$ , where  $q = 1, 2, 3, 4$ ;
6: Initialize  $FE \leftarrow L$ ; {Initialize function evaluations}
7: while  $FE < MaxEval$  do
8:   Apply the DRL-based AOS mechanism to generate offspring populations  $Off_m$  and  $Off_a$ ; {See Algorithm 2}
9:    $PO_m \leftarrow P_m \cup Off_m$ ;
10:   $PO_a \leftarrow P_a \cup Off_a$ ;
11:  Evaluate individuals in  $PO_m$  on the main task and in  $PO_a$  on the auxiliary tasks;
12:  Generate knowledge transfer individuals  $v_{a1}^*$  and  $v_{m1}^*$ ; {See Algorithm 3}
13:   $P'_m \leftarrow PO_m \cup v_{a1}^*$ ;
14:   $P'_a \leftarrow PO_a \cup v_{m1}^*$ ;
15:   $P_m \leftarrow$  Select  $L$  individuals for  $P'_m$  using the CDP method;
16:   $P_a \leftarrow$  Select  $L$  individuals for  $P'_a$  using the  $\varepsilon$ -constraint technique;
17:   $FE \leftarrow FE + L$ ; {Update function evaluations}
18: end while
19: return The non-dominated solutions from  $P_m$ .

```

4.2. The EMT framework

The proposed EMT framework for addressing MOMUPP is illustrated in Fig. 8. Within this environment, MOMUPP serves as the main task,

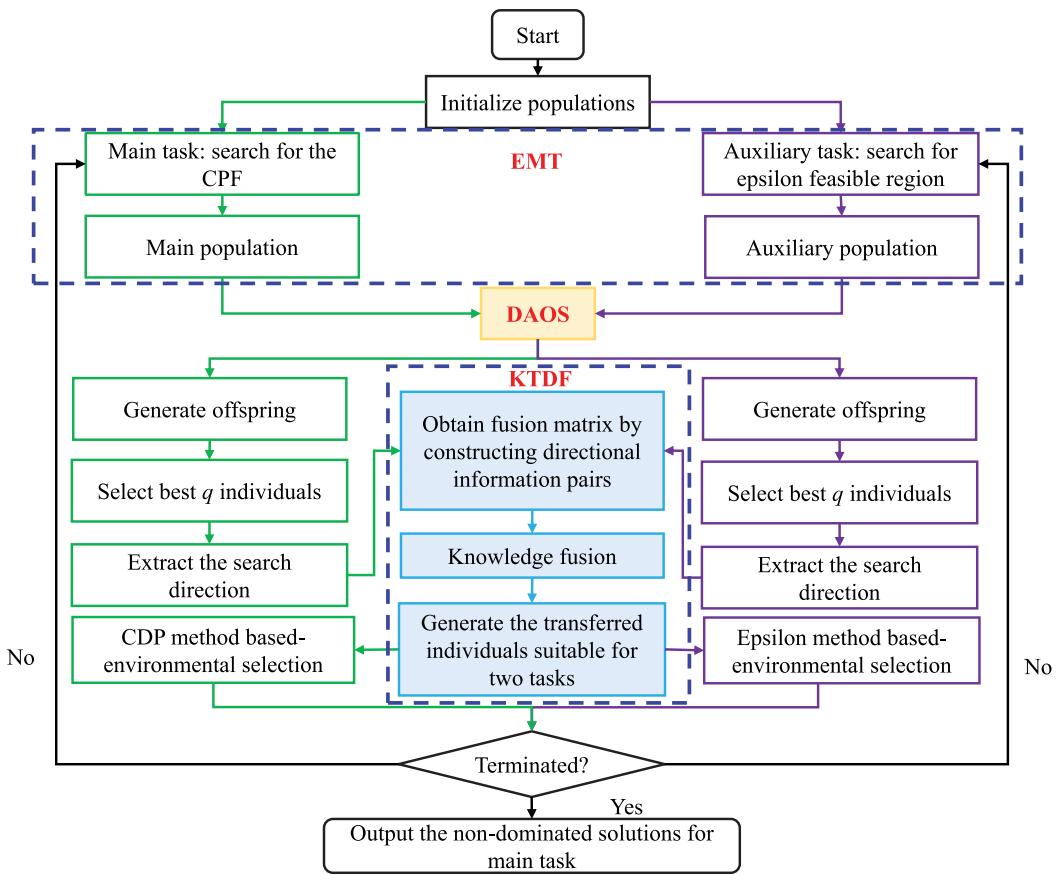


Fig. 7. The flowchart of the proposed EMMOP.

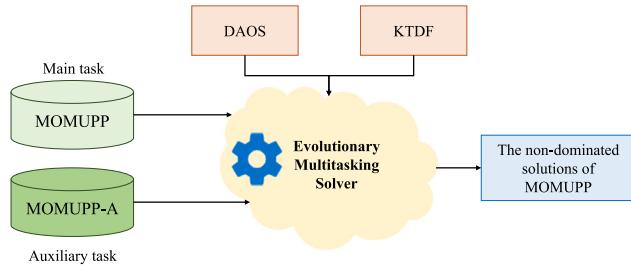


Fig. 8. Illustration of solving MOMUPP with EMT.

while a simplified variant, MOMUPP-A, is introduced as an auxiliary task to support its optimization. Although MOMUPP-A is computationally less intensive, its strong similarity to the main task enables it to guide the search process and accelerate convergence. It is important to note that, although the main and auxiliary tasks differ in their feasible solution spaces due to distinct constraint settings, they share a common decision space. Specifically, the feasible solution space is constrained by task-specific limitations, whereas the decision space is determined by the number of path points and the encoding scheme employed. Since both tasks adopt the same encoding method and an identical number of path points, they exhibit a structurally consistent representation. This consistency facilitates direct and effective knowledge transfer between the tasks, thereby enhancing the optimization performance of the main task.

4.3. DAOS-based offspring generation method

Given the unique characteristics of each task, our goal is to select the most suitable search operator for each one. To this end, the AOS

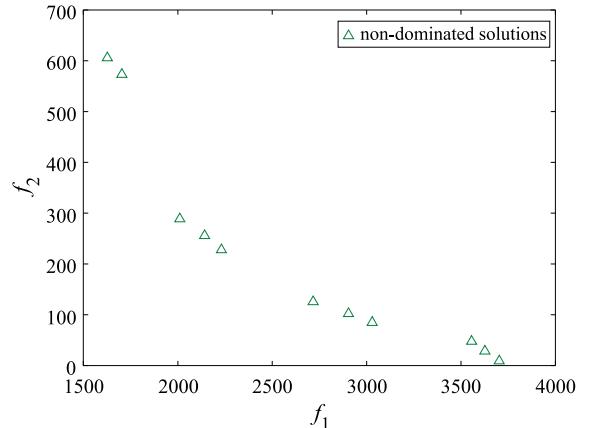


Fig. 9. Optimization results of the MOMUPP problem in a test scenario.

method is applied independently within a task population, dynamically identifying the most effective operator during optimization. Moreover, we integrate the D2QN algorithm within the DRL framework to refine AOS for the parent population, thereby promoting the generation of high-quality offspring.

In the D2QN algorithm, an agent observes the current state of the environment, chooses an action, and receives a reward. Through iterative learning, the algorithm seeks to maximize cumulative rewards. The key elements—state, action, and reward—are crucial to the D2QN framework. Their design must be specifically aligned to meet the needs of the proposed MOMUPP problem. The following subsections provide a detailed explanation of these components.

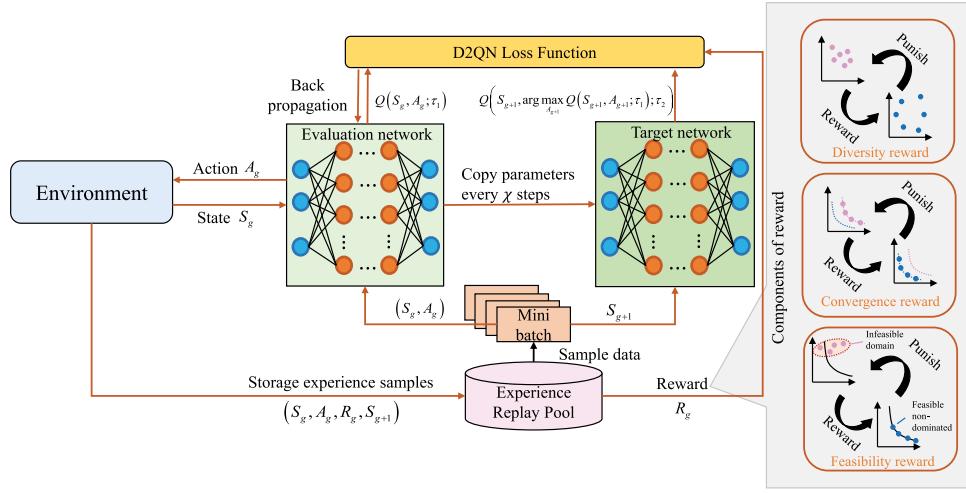


Fig. 10. The D2QN training process.

4.3.1. State, action, and reward

(1) State space: The construction of the state space is critical for ensuring the reliability and performance of the Q-network. To accurately represent the current state of the population, this study selects ten distinct state features for the D2QN. These features are categorized into four groups: path length, threat cost, feasibility, and diversity. Specifically, three features capture the maximum, minimum, and mean values of path length, while three analogous features are used for threat cost. Feasibility is evaluated through the average constraint violation and the proportion of feasible solutions in the population. Diversity is assessed based on the distribution of solutions in the objective space and their mean distances in the decision space. The state features are defined as follows:

$$s_1 = (f_1^{\text{best}}, f_2^{\text{best}}), \quad (21)$$

$$s_2 = (f_1^{\text{worst}}, f_2^{\text{worst}}), \quad (22)$$

$$s_3 = (f_1^{\text{ave}}, f_2^{\text{ave}}), \quad (23)$$

$$s_4 = \frac{\sum_{q=1}^Y \max \{0, C_q(W)\}}{Y}, \quad (24)$$

$$s_5 = \# \text{Num} \left(\sum_{q=1}^Y \max \{(0, C_q(W))\} = 0 \right) / L, \quad (25)$$

$$s_6 = \sqrt{\sum_{i=1}^L \frac{(D_i - \frac{D_{l-1}}{L-1})^2}{L-1}}, \quad (26)$$

$$D_i = \min_{W_i \in L, W_u \neq W_i} \sum_{b=1}^2 |F_b(W_i) - F_b(W_u)|, \\ s_7 = \frac{\sum_{i=1}^L \sum_{u=i+1}^L d_{iu}}{L(L-1)/2}, \quad (27)$$

where f_b^{best} , f_b^{worst} , and f_b^{ave} denote the best, worst, and mean values of the cost functions f_b , respectively, with $b \in \{1, 2\}$. The term $\# \text{Num} \left(\sum_{q=1}^Y \max \{(0, C_q(W))\} = 0 \right)$ stands for the number of feasible solutions, L is the total number of solutions, and d_{iu} denotes the Euclidean distance of two solutions.

To reduce the influence of different scenarios on the network, all parameters related to path length and threat are standardized. Consequently, the state space is constructed as follows:

$$S_g = \{s \mid s = (s_1, s_2, s_3, s_4, s_5, s_6, s_7)\}. \quad (28)$$

(2) Action space: Traditional path planning algorithms often assume that optimal solutions in the objective space are continuous and uniformly distributed, leading to the selection of either the optimal or a random individual for mutation. However, this assumption fails in battlefield environments, where the spatial distribution is complex, and multiple feasible paths may have similar flight lengths and safety levels. As shown in Fig. 9, the non-dominated solutions in these environments is non-uniform, which can result in local optima and poor convergence when traditional algorithms are used.

To tackle this problem, we introduce an improved mutation operator centered on the population centroid. In battlefield environments, flight paths with similar performance often exhibit comparable spatial structural features, particularly in terms of their flight environments. Therefore, during the evolutionary process, individuals sharing similar flight environment characteristics should be prioritized. To achieve this, the population centroid is introduced as a constraint in the optimization process, with the centroid vector capturing these shared structural features. The population centroid $X_{cen,g}$ is defined as:

$$X_{cen,g} = \frac{\sum_{l=1}^L X_{l,g}}{L}, \quad (29)$$

Based on the population centroid, we propose two enhanced mutation operators as follows:

$$U_{l,g}^1 = X_{l,g} + F \times (X_{cen,g} - X_{l,g}) \\ + F \times (X_{a_1,g} - X_{a_2,g}), \quad (30)$$

$$U_{l,g}^2 = X_{a_1,g} + F \times (X_{cen,g} - X_{l,g}) \\ + F \times (X_{a_2,g} - X_{a_3,g}), \quad (31)$$

where $U_{l,g}$ denotes the mutant vector, F is a scaling factor, and a_1, a_2, a_3 are three different random integers between 1 and L , distinct from index l .

To further refine the action space, we introduce three additional operators: DE/rand/1 [62], SBX [63], and competitive swarm optimize (CSO) [64]. DE/rand/1 and SBX help maintain population diversity, with SBX particularly effective in multi-modal scenarios. CSO strikes a balance between convergence and diversity, performing well in large-scale optimization problems. These operators are defined as follows:

- DE/rand/1

$$U_{l,g}^3 = X_{a_1,g} + F \times (X_{a_2,g} - X_{a_3,g}), \quad (32)$$

- SBX

$$U_{l,g}^4 = 0.5 \times [(X_{a_1,g} + X_{a_2,g}) + \theta (X_{a_1,g} - X_{a_2,g})], \quad (33)$$

$$\varpi = \begin{cases} (2\varpi)^{\frac{1}{v+1}}, & \text{if } \varpi \leq 0.5 \\ (2 - 2\varpi)^{\frac{-1}{v+1}}, & \text{if } \varpi > 0.5, \end{cases} \quad (34)$$

• CSO

$$U_{l,g}^5 = X_{e,g} + r_1 V_{l,g} + r_2 (X_{w,g} - X_{e,g}) + \lambda r_3 (\bar{X}_g - X_{e,g}), \quad (35)$$

where ϖ stands for a random number within $(0,1)$, and v denotes the adjustable parameter. $V_{e,g}$ indicates the velocity vector of the loser in the g th iteration. \bar{X}_g represents the average location in the g th iteration, with λ controlling the influence of \bar{X}_g . $X_{w,g}$ and $X_{e,g}$ indicate the locations of the winner and the loser in the g th iteration, respectively.

In summary, the action space is defined as follows:

$$A_g = \{U_{l,g}^1, U_{l,g}^2, U_{l,g}^3, U_{l,g}^4, U_{l,g}^5\}. \quad (36)$$

(3) Reward function: The reward function plays a key role in directing the agent's learning process. After executing an action, the agent receives a reward, which updates its strategy to improve performance in subsequent iterations. In this paper, the reward function is structured around three fundamental aspects:

- Convergence: Measured by the fitness improvement of the objective functions.
- Diversity: Assessed using the hypervolume (HV) indicator.
- Feasibility: Evaluated based on the total constraint violation degree.

Specifically, four reward levels are defined based on the fitness improvement of the two objectives (path length and threat):

- If both objectives improve (i.e., both path length and threat decrease), the reward is set to 1.5.
- If one objective improves and the improvement outweighs the degradation of the other objective, the reward is 0.1.
- If one objective improves, but the improvement is smaller than the degradation of the other objective, the reward is 0.05.
- If neither objective improves, the reward is -0.35.

Thus, the reward R_g^I for convergence is defined as:

$$R_g^I = \begin{cases} 1.5, & \text{if } \Delta f_1 \Delta f_2 > 0 \text{ and } \Delta f_1 + \Delta f_2 < 0 \\ 0.1, & \text{if } \Delta f_1 \Delta f_2 < 0 \text{ and } \Delta f_1 + \Delta f_2 < 0 \\ 0.05, & \text{if } \Delta f_1 \Delta f_2 < 0 \text{ and } \Delta f_1 + \Delta f_2 > 0 \\ -0.35, & \text{if } \Delta f_1 \Delta f_2 > 0 \text{ and } \Delta f_1 + \Delta f_2 > 0, \end{cases} \quad (37)$$

where $\Delta f_b = f_b(X_{i,g}) - f_b(X_{i,g-1})$ represents the change in objective b between iterations g and $g-1$.

The reward R_g^{II} for diversity is calculated as the relative improvement in HV:

$$R_g^{II} = \frac{HV_g - HV_{g-1}}{HV_{g-1}}, \quad (38)$$

where HV_g denotes the hypervolume at iteration g .

The reward R_g^{III} for feasibility is based on the reduction in constraint violation:

$$R_g^{III} = \frac{\phi_g - \phi_{g-1}}{\phi_{g-1}}, \quad (39)$$

where ϕ_g indicates the total constraint violation degree at iteration g .

The overall reward is a weighted combination of the three components:

$$R_g = (1 - \rho) \times R_g^{III} + \rho \times [\delta \times R_g^{II} + (1 - \delta) \times R_g^I], \quad (40)$$

where ρ is the feasibility ratio, and δ refers to the ratio of individuals that improved after applying search operators to the total population.

4.3.2. D2QN design and training process

D2QN consists of two Q-networks: the evaluation network $Q_E(\tau_1)$ and the target network $Q_T(\tau_2)$, both of which share the same six-layer architecture. The network structure includes an input layer, four fully connected (FC) layers, and an output layer. Initially, a tensor with ten state features is input into the network. This tensor passes through the FC layers, which consist of 48, 54, 54, and 48 neurons, respectively. A Sigmoid activation function is applied to each FC layer. The output layer consists of five units, each representing one of the five possible actions.

Before implementing the D2QN, the neural network must undergo training. The schematic diagram illustrating the D2QN training process is depicted in Fig. 10. Initially, a random batch of samples is selected from the experience replay pool. Subsequently, the target Q-value is calculated as follows:

$$\hat{Q}_g = R_g + \gamma \max_{A_{g+1}} Q(S_{g+1}, A_{g+1}; \tau_1); \tau_2, \quad (41)$$

where $\gamma \in [0, 1]$ denotes the discount factor, τ_1 and τ_2 represent the weights of the evaluation and target networks, respectively, and $\arg \max_{A_{g+1}} Q(S_{g+1}, A_{g+1}; \tau_1)$ chooses the action yielding the maximum Q-value in $Q_E(\tau_1)$. The target Q-value \hat{Q}_g is then utilized to update the weights of $Q_E(\tau_1)$ via gradient descent on the defined loss function:

$$L(\tau_1) = \frac{1}{b} \sum_{g \in b} [\hat{Q}_g - Q(S_g, A_g; \tau_1)]^2, \quad (42)$$

where b denotes the batch size. If the number of network iterations surpass the threshold χ , τ_2 is replaced by τ_1 .

4.3.3. Implementation of offspring generation based on DAOS

Algorithm 2 presents the pseudocode for offspring generation within the DAOS framework. The procedure begins with the initialization of an empty experience replay pool (ER) and the calculation of the initial state S_g . The main loop continues iterating until a predefined termination criterion is met. The evolutionary process consists of two distinct stages. During the initial stage, random actions are selected to collect sufficient historical data for training the D2QN model. In the second stage, actions are chosen using an ϵ -greedy strategy, which balances exploration and exploitation. The selected actions are applied to the current population to generate offspring, while the corresponding rewards R_g and subsequent states S_{g+1} are computed. Each transition (S_g, A_g, R_g, S_{g+1}) is then stored in ER , enabling iterative refinement of the D2QN model through continuous learning.

4.4. KTDF mechanism

After generating the offspring population, the EMMOP framework initiates knowledge transfer between the main and auxiliary tasks. An effective knowledge transfer mechanism is essential for improving search performance, accelerating convergence, and reducing computation time. To this end, we propose the KTDF mechanism to enhance transfer efficiency. Specifically, the mechanism first extracts the search directions of elite individuals from both the main and auxiliary task populations. These direction vectors are then fused to form a unified direction matrix. The covariance matrix is subsequently constructed based on this matrix and subjected to eigenvalue decomposition to obtain the eigenvectors. The eigenvector corresponding to the largest eigenvalue is selected to construct a knowledge sharing matrix, which is then used to guide cross-task knowledge transfer. An illustrative example of the KTDF mechanism is shown in Fig. 11.

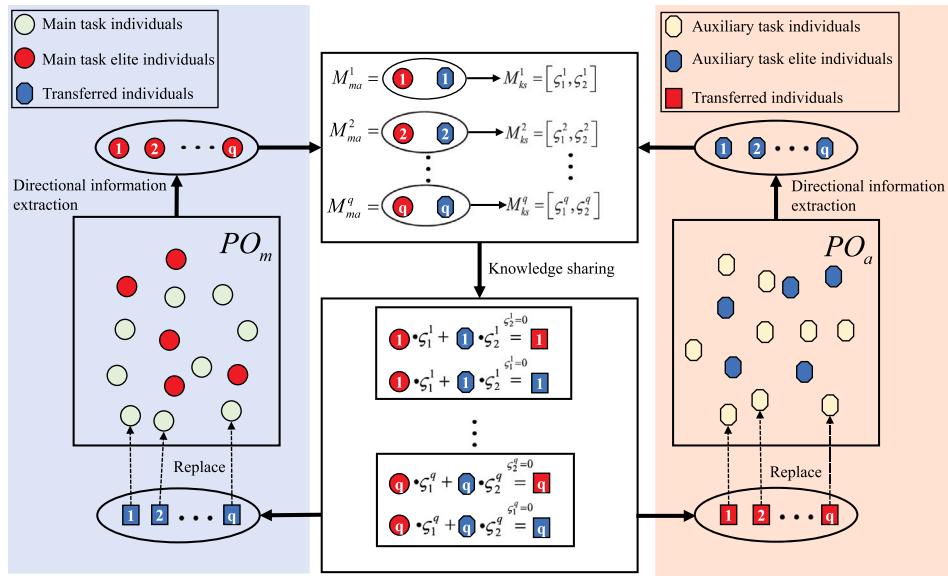


Fig. 11. Core process of KTDF.

Algorithm 2 Procedure of offspring generation based on DAOS

Input: Population size (L), candidate operator set (A_g), maximum number of iterations (g_{max}), and D2QN parameters.
Output: Offspring population Off .

- 1: Initialize the experience replay pool ER ;
- 2: Build the D2QN using Eq. (41);
- 3: $S_g \leftarrow$ Calculate the state of the population;
- 4: **while** no termination criterion is met **do**
- 5: **if** $g \leq 0.6 \times g_{max}$ **then**
- 6: Choose an action randomly from A_g ;
- 7: **else**
- 8: Choose the action using ϵ -greedy strategy based on D2QN;
- 9: **end if**
- 10: Generate offspring population Off by executing the selected action;
- 11: Obtain the reward R_g and compute the new state S_{g+1} ;
- 12: Add the transition (S_g, A_g, R_g, S_{g+1}) to ER ;
- 13: **if** update network **then**
- 14: Sample a batch of transitions (S_g, A_g, R_g, S_{g+1}) from ER randomly;
- 15: Update the evaluation network according to Eq. (42);
- 16: **end if**
- 17: **if** target network has been updated for χ iterations **then**
- 18: Update the target network: $\tau_2 \leftarrow \tau_1$;
- 19: **end if**
- 20: **end while**
- 21: **return** Off .

4.4.1. Directional information extraction

Since it is challenging to identify useful knowledge without prior information, some studies choose to preserve both the parent and offspring populations, as each may contain valuable search information [65]. Compared to knowledge transfer based directly on solution vectors, direction vectors encode the local search behavior of individuals, capturing how solutions evolve rather than merely indicating their positions. This dynamic representation more accurately reflects underlying search tendencies, reduces sensitivity to task-specific coordinate systems, and facilitates more robust and generalizable knowledge transfer across tasks.

To define individual search directions, we adopt the fitness evaluation strategy from SPEA2 [66] to select the top q individuals with the highest fitness values. Let h denote the fitness rank of an individual, and let $Z_{h,g}^t$ represent the index of the h th ranked individual in generation g for task t . The search direction of the elite individual, denoted as O_h^t , is then defined as:

$$O_h^t = X_{Z_{h,g}^t}^t - X_{Z_{h,g}^t, g-1}^t, \quad (43)$$

where $X_{Z_{h,g}^t}^t$ and $X_{Z_{h,g}^t, g-1}^t$ denote the solution of the same individual in the current and previous generations, respectively. This formulation captures the movement direction of each elite individual across generations, thereby reflecting its localized search behavior.

4.4.2. Knowledge fusion

In this study, the directional information derived from parent and offspring individuals is regarded as transferable knowledge. Effectively leveraging this information across tasks remains a significant challenge. Unlike traditional approaches that directly transfer knowledge from the auxiliary task to the main task, our method first fuses directional information from both tasks prior to transfer, thereby enhancing the quality and relevance of the knowledge being transferred.

Let $B_m = \{O_1^1, \dots, O_q^1\}$ and $B_a = \{O_1^2, \dots, O_q^2\}$ denote the sets of directional vectors extracted from the populations of the main and auxiliary tasks, respectively. Directional vectors from these sets are paired to construct fusion matrices. Taking the first pair as an example, the corresponding fusion matrix M_{ma}^1 is defined as:

$$M_{ma}^1 = \begin{pmatrix} O_1^1 \\ O_2^1 \end{pmatrix}. \quad (44)$$

Next, a covariance matrix is constructed to capture the correlation of directional movements between the two tasks:

$$M_c^1 = \frac{1}{D} M_{ma}^1 (M_{ma}^1)^T, \quad (45)$$

where D is the dimensionality of decision variables. Eigenvalue decomposition is then performed on M_c^1 :

$$(J_c^1)^T M_c^1 J_c^1 = \Lambda^1 = \text{diag}(\Omega_1, \Omega_2), \quad (46)$$

where J_c^1 contains the eigenvectors of M_c^1 , and Λ^1 is a diagonal matrix with eigenvalues $\{\Omega_1, \Omega_2\}$. The eigenvector associated with the

largest eigenvalue represents the most significant shared directional component and is selected to construct the knowledge-sharing matrix:

$$M_{ks}^1 = \text{Sort}(J_c^1) = (\zeta_1^1, \zeta_1^2), \quad (47)$$

where $\text{Sort}(\cdot)$ denotes sorting in descending order of eigenvalues. The elements ζ_1^1 and ζ_1^2 act as share factors for individual knowledge transfer between O_1^1 and O_1^2 .

Using Eqs. (44) and (47), the final knowledge transfer equations are:

$$v_{a1}^* = O_1^1 \times \zeta_1^1, \quad (48)$$

$$v_{m1}^* = O_1^2 \times \zeta_1^2, \quad (49)$$

where v_{a1}^* and v_{m1}^* denote the transferred knowledge applied to the main task and auxiliary task, respectively. The pseudocode of the proposed KTDF mechanism is provided in Algorithm 3.

Algorithm 3 Pseudocode of the KTDF Mechanism

Input: PO_m and PO_a (the merged populations of the main and auxiliary tasks, respectively.)
Output: v_{m1}^* and v_{a1}^* (the transformed individuals for the main and auxiliary tasks, respectively.)

- 1: Select the top q individuals from PO_m and PO_a based on the fitness evaluation strategy;
- 2: Construct directional vector sets B_m and B_a for the main and auxiliary tasks, respectively;
- 3: **for** $i = 1$ **to** q **do**
- 4: Compute the search direction for each elite individual using Eq. (43);
- 5: Construct the knowledge-sharing matrix using Eqs. (44)–(47);
- 6: Generate the transformed individuals v_{m1}^* and v_{a1}^* via knowledge transfer using Eqs. (48) and (49);
- 7: **end for**
- 8: **return** v_{m1}^* and v_{a1}^* ;

4.5. Computational complexity analysis

The computational overhead of EMMOP is primarily determined by four key components: the DAOS mechanism, offspring generation, the KTDF mechanism, and environmental selection. The time complexity of the DAOS mechanism is $O\left(\frac{1}{2}L^2\right)$ for state calculation, while the neural network training process, utilizing backpropagation, incurs a time complexity of $O(g_{max} \cdot G^2)$. Offspring generation has a complexity of $O(L \cdot G)$. The KTDF mechanism has an overall complexity of $O(2 \cdot q^2 \cdot G)$, where the complexity of constructing the knowledge-sharing matrices $O(q \cdot G)$, and the generation of knowledge transfer individuals is $O(2 \cdot q)$. Finally, the environmental selection for both the main and auxiliary tasks contributes a computational complexity of $O(t \cdot L^2)$, where t is the number of objectives. In summary, the overall computational complexity of EMMOP is dominated by the term $O(2 \cdot g_{max} \cdot G^2)$, which reflects the main computational cost, as the other terms are comparatively smaller in magnitude.

The space complexity of the EMMOP is primarily influenced by population size and the size of the neural network used in DAOS. Specifically, population size incurs a space complexity of $O(2 \cdot L \cdot G)$, while storing the weights of the D2QN network contributes a space complexity of $O(4 \cdot ne^2 \cdot wi)$, where ne denotes the network depth and wi is proportional to the width of each layer.

5. Simulation results and analysis

This section begins by outlining the experimental setup, which includes test cases, comparative methods, parameter settings, and performance metrics. Subsequently, we conduct orthogonal experiments to examine the key parameters of EMMOP. Following that, we evaluate

and compare the performance of various algorithms, and then present an in-depth investigation of EMMOP across multiple scenarios. In addition, we perform ablation studies to assess the effectiveness of the EMT framework and DAOS strategy. Finally, we present the real flight results obtained from experiments on the V-REP platform.

5.1. Experimental setup

Since no established benchmark exists for MOMUPP, we construct six representative mountain environments using data retrieved from the Google Maps API. Based on these maps, we generate fifteen test cases, which are summarized in Tables S1 and S2 (supplementary materials). Each scenario incorporates key threat-related parameters relevant to UAV path planning, including the number of UAVs, threat center locations, radii, heights, and the starting and target positions.

For comparative analysis, we evaluate the proposed EMMOP against five state-of-the-art algorithms: Improved MOEA/D [11], INSGA-II [15], DSNSGA-III [13], LCMOEA [67], and DRLOS-EMCMO [68]. Among them, Improved MOEA/D, INSGA-II, and DSNSGA-III were proposed in the past three years specifically for addressing MOMUPP problems. LCMOEA integrates deep learning techniques to enhance the effectiveness of evolution and has demonstrated competitive performance on high-dimensional CMOPs. DRLOS-EMCMO, a leading CMOEA, significantly influenced the design of EMMOP by employing operator selection mechanisms for offspring generation. Each algorithm is executed independently 30 times per scenario, with each run consisting of 120,000 function evaluations and a population size of 60 individuals.

To evaluate algorithmic performance, we adopt two widely used metrics: hypervolume (HV) and pure diversity (PD). A higher HV indicates better overall convergence and distribution, while a higher PD reflects greater diversity in the approximated PF. To further assess performance differences, the enhancement percentage (EP) [69] is introduced, which quantifies the relative improvement of our method over the compared algorithms. It is defined as:

$$\text{EP}(Z) = \frac{Z(RE_{Our}) - Z(RE_{Compared})}{Z(RE_{Compared})} \times 100\%, \quad (50)$$

where $Z(RE_{Our})$ and $Z(RE_{Compared})$ denote the performance of our method and the compared method, respectively, on metric Z . A positive EP indicates that EMMOP outperforms the baseline in the given metric, with larger values representing more significant improvements. Conversely, a negative EP suggests inferior performance.

In addition to EP, we employ the relative performance deviation (RPD) [70] metric to measure the deviation of a given algorithm from the best-known solution. It is formulated as:

$$\text{RPD} = \frac{F_b - F_c}{F_b} \times 100\%, \quad (51)$$

where F_b is the best value achieved across all algorithms, and F_c is the best result obtained by an algorithm c . Based on this, we calculate the mean RPD (mRPD) and best RPD (bRPD) across all test instances. Lower values of mRPD and bRPD indicate better performance.

5.2. Parameters analysis

The EMMOP is optimized by tuning six parameters: learning rate (lr), discount factor (γ), greedy factor (ϵ), batch size (b), target network update interval (χ), and population size (K). To facilitate parameter optimization, the Taguchi method is implemented within a design-of-experiment framework. The three levels for each parameter are defined as follows: $lr = \{0.001, 0.005, 0.01\}$, $\gamma = \{0.9, 0.93, 0.95\}$, $\epsilon = \{0.8, 0.9, 0.95\}$, $b = \{32, 64, 128\}$, $\chi = \{6, 8, 10\}$, and $K = \{40, 50, 60\}$. The orthogonal array $L_{27}(3^6)$, as presented in Table S3 (supplementary materials), is utilized to systematically investigate combinations of parameter settings. Each configuration is independently evaluated 15

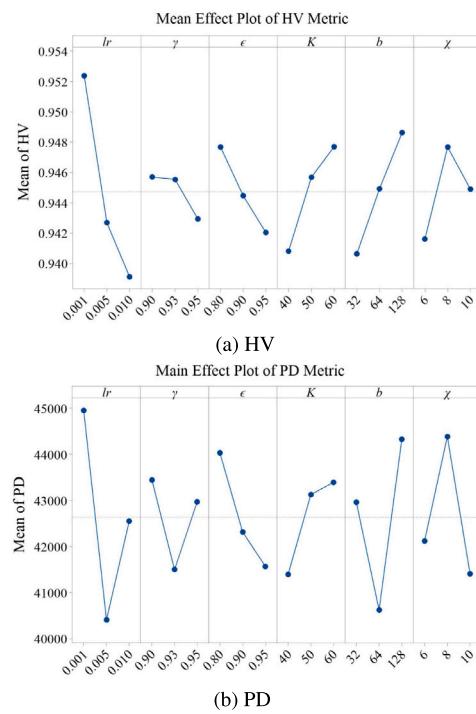


Fig. 12. Main factor effects plots of all metrics.

times on randomly generated test instances, and the average performance metrics are recorded. Based on the results in Table S3, the factor-level effects are depicted in Fig. 12. As shown in Fig. 12, the optimal parameter values are set as follows: $lr = 0.001$, $\gamma = 0.9$, $\epsilon = 0.8$, $K = 60$, $b = 128$, and $\chi = 8$.

5.3. Comparison experiments and analysis

Tables 2 and 3 present the experimental results, reporting the mean and standard deviation (Std) values of the HV and PD metrics for all compared algorithms across fifteen test cases. The best value for each test case is highlighted in bold. Additionally, the Wilcoxon rank-sum test at a 95% confidence level is applied to assess the statistical significance of performance differences. The symbols “+/-/-” indicate whether the proposed EMMOP outperforms, is comparable to, or underperforms relative to a baseline method, respectively. EP values in the last column are calculated according to Eq. (50), where $Z(Re_{Our})$ and $Z(Re_{Compared})$ denote the HV (or PD) values obtained by EMMOP and the best competing algorithm, respectively. The symbol ‘‡’ marks the best HV or PD value among the baselines.

From the numerical results in Tables 2 and 3, EMMOP consistently outperforms all compared algorithms in terms of both HV and PD on nearly all test cases. Specifically, according to the Wilcoxon rank-sum test, EMMOP achieves significantly better HV values than INSGA-II, DSNSGA-III, Improved MOEA/D, LCMOEAs, and DRLOS-EMCMO on 15, 15, 13, 15, and 14 test instances, respectively. This demonstrates its strong capability in maintaining both convergence and solution diversity. Regarding the PD metric, EMMOP performs worse than the competitors in only one case, further confirming its advantage in preserving population diversity. Furthermore, the average EP values show that EMMOP achieves 2.46% improvement in HV and 28.27% improvement in PD compared to the best-performing baseline algorithm, further validating its superior overall performance. The statistical results for the RPD metric are provided in Tables S4 and S5 (supplementary materials). As shown in Table S4, EMMOP attains the best performance in terms of both mRPD and bRPD in nearly all test cases, achieving

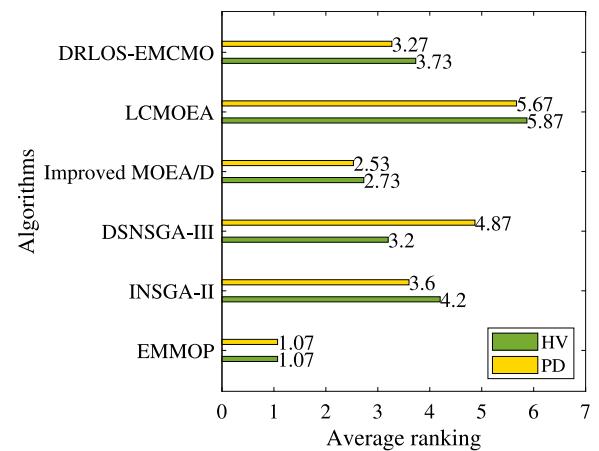


Fig. 13. Average rankings of all compared algorithms based on the Friedman test.

an average mRPD improvement of 3%–9% compared to competing algorithms. Similarly, Table S5 shows that EMMOP outperforms all five baseline methods with an average mRPD improvement ranging from 23% to 55%.

Table 4 further presents results of a multi-problem Wilcoxon signed-rank test [71]. The statistical results indicate that all p -values are below 0.05, confirming the presence of significant performance differences between EMMOP and the compared methods. Moreover, in all pairwise comparisons, the R^+ values exceed the R^- values, providing additional evidence of EMMOP's superiority. To assess overall algorithmic performance across the 15 test problems, the average HV and PD values of the six algorithms were ranked using the Friedman test, as illustrated in Fig. 13. EMMOP attains the best average rank of 1.07 on both metrics, underscoring its strong and consistent optimization capability.

To provide a more intuitive comparison, Fig. 14 displays the approximate PFs generated by all the compared algorithms for the six representative scenarios: Map 2 (2 UAVs), Map 2 (4 UAVs), Map 4 (2 UAVs), Map 4 (3 UAVs), Map 5 (3 UAVs), and Map 5 (4 UAVs). In these figures, the solutions obtained by EMMOP are predominantly clustered closer to the lower left corner, indicating superior convergence and diversity. This demonstrates that the proposed EMMOP yields path planning solutions with both shorter total path lengths and reduced threat exposure.

Fig. 15 further illustrates the HV convergence curves for all algorithms across several scenarios, including Map 2 (2 UAVs), Map 4 (4 UAVs), Map 5 (3 UAVs), and Map 5 (4 UAVs), providing a clear depiction of their evolutionary processes. The horizontal axis represents the iteration number while the vertical axis represents the HV. Although the number of UAVs that Map 2 (2 UAVs) and Map 5 (3 UAVs) needs to handle is relatively small, we find that the proposed algorithm works slightly better than INSGA-II, DSNSGA-III, the Improved MOEA/D, LCMOEAs, and DRLOS-EMCMO. The advantage of the proposed algorithm is further shown by the results on Map 4 (4 UAVs) and Map 5 (4 UAVs), where more UAVs are required to be planned, and the constraints in Eq. (17) are more difficult to satisfy compared with Map 2 (2 UAVs) and Map 5 (3 UAVs). EMMOP converges rapidly and consistently outperforms its competitors in solution quality, effectively guiding the population toward the approximated PF. These results underscore that EMMOP achieves higher-quality solutions for the MOMUPP problem.

Fig. 16 and Table 5 present the comparative results of all algorithms under various preferences on the Map 5 (3 UAVs). For preference 1, which primarily emphasizes the optimal path length, EMMOP achieves an optimal solution of 4802.83, reflecting improvements of 21.80%, 60.09%, 46.88%, 62.08%, and 7.88% compared to INSGA-II, DSNSGA-III, Improved MOEA/D, LCMOEAs, and DRLOS-EMCMO respectively.

Table 2The HV results (Mean \pm Std) obtained from all comparison algorithms.

Scenarios	The number of UAVs	EMMOP	INSGA-II	DSNSGA-III	Improved MOEA/D	LCMOEA	DRLOS-EMCMO	EP(HV)
Map1	2 UAVs	0.9249 \pm 0.0213	0.8616 \pm 0.0649	0.8722 \pm 0.0607	0.8985 \pm 0.0450 \ddagger	0.7668 \pm 0.0852	0.8186 \pm 0.0536	2.94%
	3 UAVs	0.8627 \pm 0.0360	0.8186 \pm 0.0330	0.8197 \pm 0.0450	0.8378 \pm 0.0429	0.7249 \pm 0.0457	0.8458 \pm 0.0304 \ddagger	2.00%
	4 UAVs	0.9075 \pm 0.0306	0.8385 \pm 0.0165	0.8625 \pm 0.0324	0.8717 \pm 0.0207 \ddagger	0.7788 \pm 0.0301	0.8197 \pm 0.0262	4.11%
Map2	2 UAVs	0.9021 \pm 0.0233	0.8741 \pm 0.0325	0.8784 \pm 0.0388	0.8803 \pm 0.0341 \ddagger	0.8155 \pm 0.0291	0.8724 \pm 0.0212	2.48%
	3 UAVs	0.8911 \pm 0.0299	0.8700 \pm 0.0354	0.8574 \pm 0.0316	0.8708 \pm 0.0415 \ddagger	0.7728 \pm 0.0494	0.8659 \pm 0.0448	2.33%
	4 UAVs	0.8743 \pm 0.0455	0.8604 \pm 0.0464	0.8532 \pm 0.0585	0.8614 \pm 0.0488 \ddagger	0.8022 \pm 0.0573	0.8399 \pm 0.0393	1.50%
Map3	2 UAVs	0.8887 \pm 0.0355	0.8470 \pm 0.0678	0.8776 \pm 0.0501 \ddagger	0.8539 \pm 0.0408	0.8060 \pm 0.0406	0.8396 \pm 0.0508	1.27%
	3 UAVs	0.8858 \pm 0.0465	0.8337 \pm 0.0298	0.8581 \pm 0.0224	0.8896 \pm 0.0442 \ddagger	0.7901 \pm 0.0603	0.8392 \pm 0.0408	-0.43%
	4 UAVs	0.8734 \pm 0.0543	0.7969 \pm 0.0599	0.8184 \pm 0.0506	0.8415 \pm 0.0416 \ddagger	0.8156 \pm 0.0676	0.8142 \pm 0.0709	3.79%
Map4	2 UAVs	0.8890 \pm 0.0282	0.8458 \pm 0.0401	0.8536 \pm 0.0567	0.8812 \pm 0.0312 \ddagger	0.8054 \pm 0.0327	0.8547 \pm 0.0317	0.89%
	3 UAVs	0.8632 \pm 0.0373	0.7926 \pm 0.0429	0.8110 \pm 0.0353	0.7983 \pm 0.0500	0.7737 \pm 0.0470	0.8255 \pm 0.0368 \ddagger	4.57%
	4 UAVs	0.8622 \pm 0.0254	0.8073 \pm 0.0426	0.8379 \pm 0.0313	0.8206 \pm 0.0383	0.7692 \pm 0.0475	0.8387 \pm 0.0311 \ddagger	2.80%
Map5	2 UAVs	0.9457 \pm 0.0132	0.9217 \pm 0.0427	0.9197 \pm 0.0440	0.9178 \pm 0.0335	0.8962 \pm 0.0292	0.9352 \pm 0.0120 \ddagger	1.12%
	3 UAVs	0.8903 \pm 0.0461	0.8368 \pm 0.0566	0.8626 \pm 0.0554	0.8570 \pm 0.0426	0.8341 \pm 0.0408	0.8861 \pm 0.0235 \ddagger	0.47%
	4 UAVs	0.8415 \pm 0.0423	0.7831 \pm 0.0350	0.7861 \pm 0.0331 \ddagger	0.7806 \pm 0.0405	0.7694 \pm 0.0544	0.7790 \pm 0.0505	7.05%
Average	/	/	/	/	/	/	/	2.46%
-/+/-		15/0/0	15/0/0	13/2/0	15/0/0	14/1/0		

Table 3The PD results (Mean \pm Std) obtained from all comparison algorithms.

Scenarios	The number of UAVs	EMMOP	INSGA-II	DSNSGA-III	Improved MOEA/D	LCMOEA	DRLOS-EMCMO	EP
Map1	2 UAVs	2.9165E4 \pm 4.1569E3	2.6986E4 \pm 5.5883E3 \ddagger	1.9509E4 \pm 6.8937E3	2.5708E4 \pm 6.8937E3	9.4376E3 \pm 2.3828E3	2.1237E4 \pm 4.4205E3	8.07%
	3 UAVs	5.8430E4 \pm 6.8661E3	4.0817E4 \pm 8.5847E3	3.2979E4 \pm 9.0591E3	4.3119E4 \pm 9.5276E3	2.0582E4 \pm 5.2951E3	4.3908E4 \pm 7.4450E3	33.07%
	4 UAVs	6.8769E4 \pm 7.2688E3	1.8974E4 \pm 6.0356E3	1.6703E4 \pm 2.0789E3	4.0639E4 \pm 2.3737E4 \ddagger	1.7849E4 \pm 3.5128E3	2.8855E4 \pm 5.0986E3	69.20%
Map2	2 UAVs	3.7345E4 \pm 9.1964E3	3.5414E4 \pm 1.1548E4	2.7627E4 \pm 1.4177E4	3.5850E4 \pm 1.6934E4 \ddagger	1.0683E4 \pm 3.3878E3	2.1402E4 \pm 5.0221E3	4.17%
	3 UAVs	5.7976E4 \pm 9.5675E3	3.7941E4 \pm 6.9689E3	2.5836E4 \pm 8.2632E3	2.9438E4 \pm 6.5811E3	1.3939E4 \pm 6.0978E3	4.3530E4 \pm 1.4662E4 \ddagger	33.19%
	4 UAVs	7.8199E4 \pm 2.3309E4	2.8985E4 \pm 1.7635E4	2.5503E4 \pm 8.9491E3	3.9597E4 \pm 1.5137E4 \ddagger	6.9217E4 \pm 4.9072E3	3.0043E4 \pm 1.1151E4	97.49%
Map3	2 UAVs	3.4258E4 \pm 1.0273E4	2.3756E4 \pm 7.6919E3	2.7240E4 \pm 7.0715E3	2.7353E4 \pm 6.6430E3 \ddagger	1.2134E4 \pm 2.7103E3	2.4572E4 \pm 5.5771E3	25.25%
	3 UAVs	6.6564E4 \pm 9.4940E3	6.1976E4 \pm 1.2064E4	5.1084E4 \pm 8.9292E3	6.6881E4 \pm 9.4083E3 \ddagger	2.8519E4 \pm 9.9617E3	5.0563E4 \pm 1.0496E4	-0.47%
	4 UAVs	6.5987E4 \pm 1.1272E4	2.8682E4 \pm 7.8209E3	2.2066E4 \pm 6.6833E3	3.4499E4 \pm 1.7492E4	1.0827E4 \pm 4.7780E3	5.0294E4 \pm 2.0361E4 \ddagger	31.20%
Map4	2 UAVs	3.2183E4 \pm 8.6605E3	2.7455E4 \pm 7.7368E3 \ddagger	2.1406E4 \pm 6.8194E3	2.6413E4 \pm 1.1435E4	9.9461E3 \pm 2.7290E3	2.3786E4 \pm 4.0903E3	17.55%
	3 UAVs	5.2952E4 \pm 1.1475E4	1.7253E4 \pm 4.5513E3	1.4888E4 \pm 5.7370E3	2.0874E4 \pm 1.3506E4	1.7254E4 \pm 6.5126E3	4.2382E4 \pm 6.9168E3 \ddagger	24.94%
	4 UAVs	8.8826E4 \pm 1.2021E4	1.8974E4 \pm 6.0356E3	1.6703E4 \pm 2.0789E3	3.8639E4 \pm 2.3737E4	2.1465E4 \pm 8.5711E3	7.8748E4 \pm 1.6913E4 \ddagger	12.80%
Map5	2 UAVs	4.2367E4 \pm 4.4300E3	3.4997E4 \pm 6.5379E3 \ddagger	2.5774E4 \pm 6.5022E3	2.8296E4 \pm 7.6500E3	4.5090E3 \pm 8.9825E2	2.7461E4 \pm 5.1942E3	21.06%
	3 UAVs	2.5200E4 \pm 5.0638E3	2.0910E4 \pm 5.8947E3	1.3496E4 \pm 3.8326E3	2.1518E4 \pm 7.5441E3 \ddagger	5.0066E3 \pm 1.4744E3	1.5519E4 \pm 3.7427E3	20.52%
	4 UAVs	9.8716E4 \pm 5.3446E4	5.0339E4 \pm 1.3596E4	5.8863E4 \pm 2.4687E4	7.9773E4 \pm 3.9218E4 \ddagger	2.2904E4 \pm 9.7120E3	7.8300E4 \pm 1.3720E4	26.01%
Average	/	/	/	/	/	/	/	28.27%
-/+/-		15/0/0	15/0/0	14/0/1	15/0/0	15/0/0		

Table 4

Multi-problem Wilcoxon signed-rank test results of all compared algorithms.

Metric	Comparisons					
	EMMOP vs INSGA-II	EMMOP vs DSNSGA-III	EMMOP vs Improved MOEA/D	EMMOP vs LCMOEAE	EMMOP vs DRLOS-EMCMO	
HV	R^+	120	120	119	120	120
	R^-	0	0	1	0	0
	p-value	6.1035E-5	6.1035E-5	1.2207E-4	6.1035E-5	6.1035E-5
	level=0.05	Yes	Yes	Yes	Yes	Yes
PD	R^+	120	120	119	120	120
	R^-	0	0	1	0	0
	p-value	6.1035E-5	6.1035E-5	1.2207E-4	6.1035E-5	6.1035E-5
	level=0.05	Yes	Yes	Yes	Yes	Yes

Thus, when the objective is to reduce flight energy consumption, EMMOP yields shorter paths. For preference 2, which primarily emphasizes the optimal threat cost, EMMOP achieves an optimal solution of 2815.97, with improvements of 44.76%, 51.02%, 43.59%, 53.54%, and 31.17% compared to INSGA-II, DSNSGA-III, Improved MOEA/D, LCMOEAE, and DRLOS-EMCMO, respectively. Consequently, when prioritizing flight safety, EMMOP delivers enhanced performance, thereby increasing the task completion success rate. For preference 3, where the path length and threat objectives hold equal importance, EMMOP still obtains a superior solution (6448.6, 3721.24), dominating the solutions produced by its competitors.

To assess the difference in computational efficiency between EMMOP and the five baseline algorithms, a series of independent experiments are performed under consistent hardware and software configurations, with the average CPU runtime adopted as the evaluation metric. The runtime results are illustrated in Fig. 17. It can be observed that all algorithms experience increased runtime as the number of UAVs grows. For EMMOP, the primary computational overhead arises from the evaluation of individuals during the search phase. In general, although EMMOP incurs a slightly longer runtime, the gap compared to other methods remains small, indicating that EMMOP achieves a good balance between optimization performance and computational efficiency.

To explain the above results more thoroughly, we further analyze the behavior and limitations of the baseline algorithms. Improved

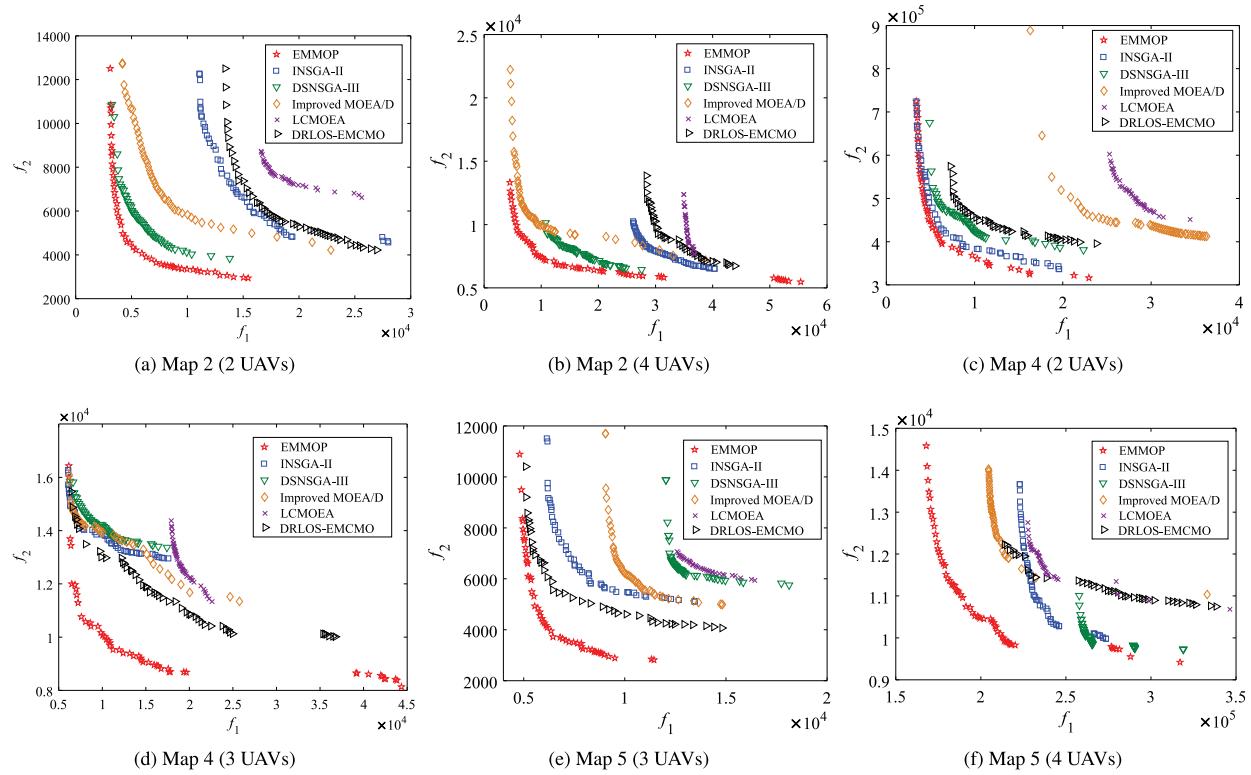


Fig. 14. Final PFs obtained by all comparison algorithms in six representative test cases.

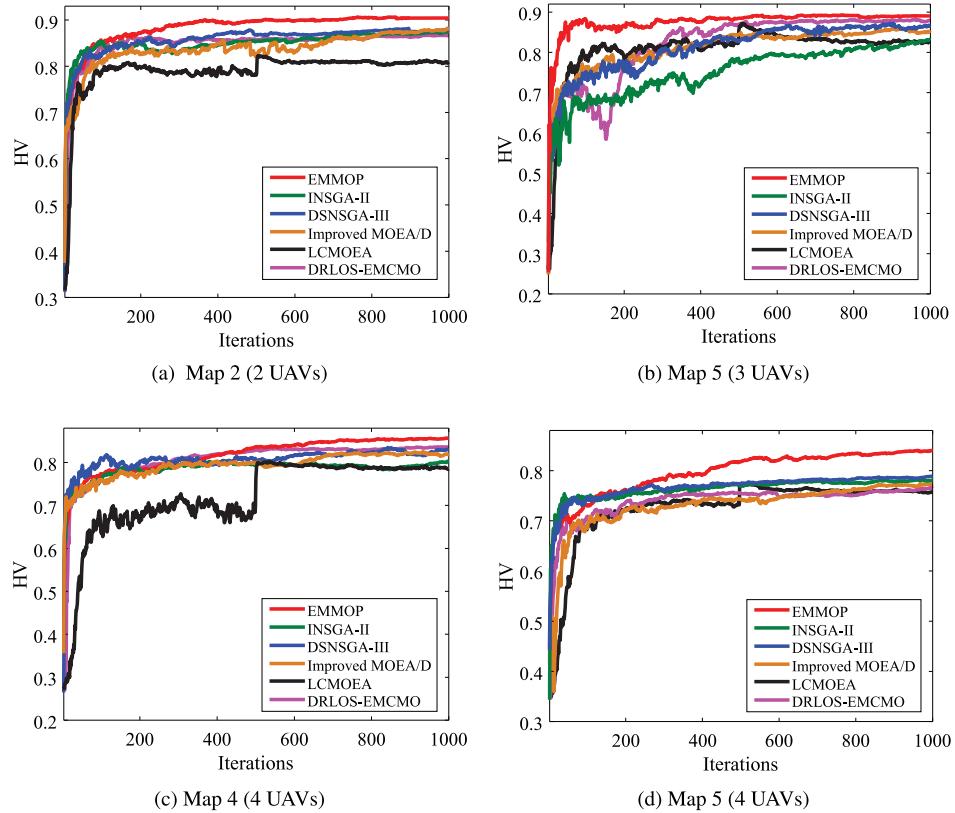


Fig. 15. Convergence plots of HV of comparison algorithms on four representative test cases.

MOEA/D performs reliably due to its decomposition-based design and dual constraint-handling strategy, yet it struggles when faced with irregular PFs in highly constrained environments. DSNSGA-III benefits

from customized crossover operators and reference-point strategies, but its performance heavily depends on the choice of reference points, limiting flexibility. INSGA-II proposes a crowding distance calculation

Table 5

The best objective values under three distinct preferences generated by the compared algorithms.

	Objective preferences	Algorithms					
		EMMOP	INSGA-II	DSNSGA-III	Improved MOEA/D	LCMOEA	DRLOS-EMCMO
Preference 1	f_1	4802.83	6141.73	12 032.70	9041.59	12 666.50	5213.69
Preference 2	f_2	2815.97	5097.68	5749.58	4992.22	6060.88	4091.41
Preference 3	f_1	6448.60	8313.92	13 020.80	10 541.80	13 410.32	6557.98
	f_2	3721.24	5817.93	6152.59	5863.67	6480.72	5630.36

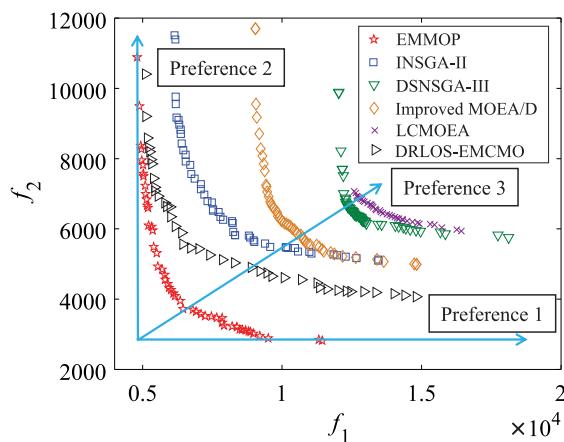


Fig. 16. Comparative figure of various algorithms with three distinct preferences.

that considers both path and objective spaces, yet the lack of guided search leads to low efficiency in constrained scenarios. DRLOS-EMCMO leverages RL to adaptively select operators, but deep networks with many parameters make it prone to overfitting under limited data, while constraint complexity further increases training costs and instability. LCMOEAE uses deep learning to guide the search via improvement vectors, but its need to train two models per generation introduces significant overhead. More importantly, in multi-UAV planning tasks with coupled constraints such as terrain occlusion, altitude limits, and threat avoidance, its learning model struggles to accurately identify feasible boundaries, which can lead to infeasible or low-diversity solutions.

In contrast, the proposed EMMOP framework achieves superior performance through the synergistic integration of three key components. First, incorporating an auxiliary population enables the main population to navigate infeasible regions more efficiently, thereby accelerating the discovery of feasible solutions and enhancing convergence. Second, the DAOS strategy effectively balances convergence and diversity among non-dominated solutions, which improves the quality of the offspring population. Finally, the KTDF strategy employs an information fusion matrix to facilitate efficient cross-task exchange of valuable information, thereby boosting both search efficiency and overall solution quality.

5.4. Performance evaluation of EMMOP in diverse maps

To further demonstrate the effectiveness of the proposed EMMOP, we extract knee point solutions from the PFs generated by EMMOP across five test cases: Map 1 (UAV4), Map 2 (UAV3), Map 3 (UAV2), Map 4 (UAV4), and Map 5 (UAV4). Fig. 18 displays the best compromise paths after solution extraction, where squares represent the start points and circles denote the corresponding target points. For each test case, the knee point is determined as the solution minimizing the distance to the ideal point. In Map 3 (UAV2) and Map 2 (UAV3), the UAVs operate under parallel and crossover modes, respectively, thereby demonstrating the algorithm's efficacy in both short- and long-distance collision avoidance. Map 1 (UAV4) further demonstrates the EMMOP's

superior performance in short-range convergence within complex environments characterized by an increasing number of UAVs and threats. In even more challenging conditions, Maps 4 (UAV4) and 5 (UAV4) feature denser threat distributions and significant terrain elevation variations. Specifically, Map 4 confirms the algorithm's capability to navigate multi-directional tasks within constrained spaces, while Map 5 demonstrates its ability to guide multiple UAVs converging from distinct areas to an adjacent region under highly complex conditions. Collectively, these results highlight the algorithm's adaptability and robustness across a wide range of environments with varying degrees of complexity.

In addition, a minimum safety distance of 10 units is enforced between UAVs to prevent collisions. The UAV positions are continuously monitored to ensure compliance with this threshold. As shown in Fig. 19, variations in inter-UAV distances during flight under the conditions illustrated in Fig. 18 consistently exceed the safety limit, effectively mitigating the risk of in-flight collisions.

To further assess the robustness of EMMOP under extreme conditions, we develop two supplementary test cases with a higher density of UAVs and threat regions. Fig. 20 displays the optimal compromise paths derived from EMMOP in these scenarios. The results indicate that EMMOP is capable of effectively addressing the MOMUPP in large-scale environments.

5.5. Ablation study

To evaluate the effectiveness of key components in EMMOP, we conduct ablation studies focusing on the EMT framework and the DAOS strategy. The detailed HV and PD results are presented in Tables 6 and 7, with comprehensive comparisons summarized by the Wilcoxon rank-sum test results reported in the last rows of both tables.

First, we evaluate the impact of the EMT framework by analyzing three modified variants of EMMOP: EMMOP1 excludes the main task, EMMOP2 removes the auxiliary task, and EMMOP3 adopts a traditional knowledge transfer strategy. Statistical comparisons of these variants are presented in Columns 3–6 of Table 6. The results show that EMMOP consistently achieves superior performance across most test cases, significantly outperforming EMMOP1, EMMOP2, and EMMOP3 on 15, 15, and 12 test instances, respectively, while performing worse on 0, 0, and only 1 instance, respectively. Similar trends are observed in the PD results shown in Table 7. Notably, both EMMOP1 and EMMOP2 demonstrate significantly weaker performance, indicating that evolving tasks in isolation — without knowledge transfer — leads to premature convergence and an inability to balance exploration and exploitation. Although EMMOP generally outperforms all variants, EMMOP3 occasionally achieves higher HV and PD values. This can be attributed to instance-specific characteristics, where a limited number of elite individuals may reduce the effectiveness of EMMOP's enhanced knowledge transfer mechanism, thereby allowing EMMOP3 to outperform under certain conditions.

Next, we evaluate the DAOS strategy using a variant termed EM-MOP4, which replaces AOS with uniform random sampling. As shown in Columns 3 and 7 of Tables 6 and 7, EMMOP outperforms EMMOP4 in most test cases, highlighting the effectiveness of AOS in adaptively selecting the optimal operator for each task at each iteration. The inferior performance of EMMOP4 is attributed to its lack of directed

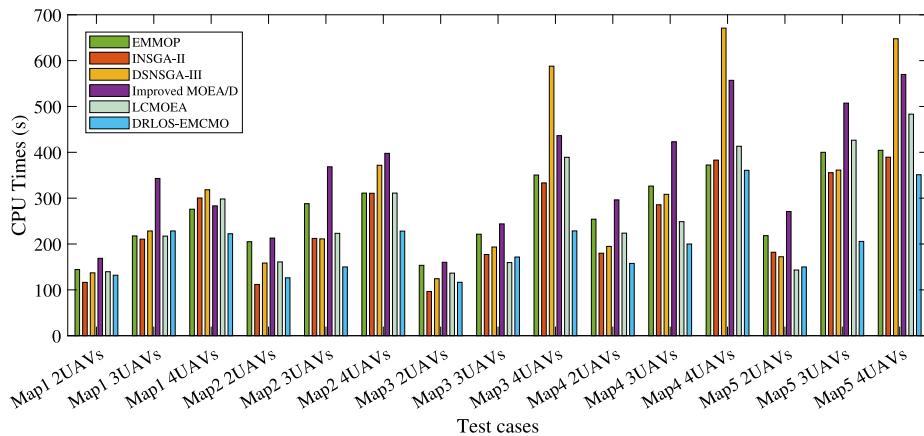


Fig. 17. Runtime comparison of algorithms in different test cases.

Table 6

The HV results (Mean \pm Std) produced by EMMOP and its four variants of degradation across all test cases.

Scenarios	The number of UAVs	EMMOP	EMMOP2	EMMOP3	EMMOP4	EMMOP5	EP (HV)
Map1	2 UAVs	0.9249 \pm 0.0213	0.8380 \pm 0.0621	0.8222 \pm 0.0751	0.8473 \pm 0.0501	0.8792 \pm 0.0513 \ddagger	5.20%
	3 UAVs	0.8627 \pm 0.0360	0.8378 \pm 0.0202 \ddagger	0.8203 \pm 0.0528	0.8362 \pm 0.0260	0.8200 \pm 0.0169	2.97%
	4 UAVs	0.9075 \pm 0.0306	0.8231 \pm 0.0257	0.8106 \pm 0.0287	0.8637 \pm 0.0268 \ddagger	0.8469 \pm 0.0354	5.07%
Map2	2 UAVs	0.9021 \pm 0.0233	0.8920 \pm 0.0263	0.8474 \pm 0.0662	0.8942 \pm 0.0307 \ddagger	0.8886 \pm 0.0325	0.88%
	3 UAVs	0.8911 \pm 0.0299	0.8555 \pm 0.0253 \ddagger	0.8417 \pm 0.0399	0.8353 \pm 0.0344	0.8273 \pm 0.0393	4.16%
	4 UAVs	0.8743 \pm 0.0455	0.8177 \pm 0.0299	0.8107 \pm 0.0254	0.8821 \pm 0.0354 \ddagger	0.7754 \pm 0.0473	-0.88%
Map3	2 UAVs	0.8887 \pm 0.0355	0.8274 \pm 0.0656	0.8541 \pm 0.0312	0.8591 \pm 0.0300 \ddagger	0.8201 \pm 0.0451	3.45%
	3 UAVs	0.8858 \pm 0.0465	0.7948 \pm 0.0449	0.8227 \pm 0.0280	0.8698 \pm 0.0195 \ddagger	0.7950 \pm 0.0193	1.84%
	4 UAVs	0.8734 \pm 0.0543	0.7640 \pm 0.0489	0.8026 \pm 0.0227	0.8763 \pm 0.0363 \ddagger	0.7536 \pm 0.0552	-0.33%
Map4	2 UAVs	0.8890 \pm 0.0282	0.8282 \pm 0.0465	0.8706 \pm 0.0383	0.8796 \pm 0.0221 \ddagger	0.8732 \pm 0.0390	1.07%
	3 UAVs	0.8632 \pm 0.0373	0.8378 \pm 0.0311	0.8100 \pm 0.0459	0.8657 \pm 0.0204 \ddagger	0.8428 \pm 0.0204	-0.29%
	4 UAVs	0.8622 \pm 0.0254	0.8346 \pm 0.0276	0.8430 \pm 0.0099	0.8564 \pm 0.0326 \ddagger	0.8331 \pm 0.0140	0.68%
Map5	2 UAVs	0.9457 \pm 0.0132	0.9288 \pm 0.0265	0.9167 \pm 0.0297	0.9356 \pm 0.0110	0.9366 \pm 0.0107 \ddagger	0.97%
	3 UAVs	0.8903 \pm 0.0461	0.8337 \pm 0.0418	0.8442 \pm 0.0431	0.8866 \pm 0.0295 \ddagger	0.8750 \pm 0.0229	0.42%
	4 UAVs	0.8415 \pm 0.0423	0.7788 \pm 0.0549	0.7913 \pm 0.0322	0.8299 \pm 0.0422 \ddagger	0.7547 \pm 0.0353	1.40%
Average	/	/	/	/	/	/	1.77%
-/+		15/0/0	15/0/0	12/2/1	15/0/0		

Table 7

The PD results (Mean \pm Std) produced by EMMOP and its four variants of degradation across all test cases.

Scenarios	The number of UAVs	EMMOP	EMMOP2	EMMOP3	EMMOP4	EMMOP5	EP (PD)
Map1	2 UAVs	2.9165E4 \pm 4.1569E3	2.0999E4 \pm 5.7380E3	2.1971E4 \pm 3.7023E3	2.0936E4 \pm 3.5324E3	2.8474E4 \pm 5.8647E3 \ddagger	32.74%
	3 UAVs	5.8430E4 \pm 6.8661E3	5.3172E4 \pm 7.6888E3	5.3641E4 \pm 1.0041E4	5.2546E4 \pm 1.1972E4	5.3855E4 \pm 9.3747E3 \ddagger	8.50%
	4 UAVs	6.8769E4 \pm 7.2688E3	2.9406E4 \pm 4.7509E3	2.9723E4 \pm 2.6537E3	3.5847E4 \pm 6.9978E3 \ddagger	3.1751E4 \pm 5.7583E3	91.84%
Map2	2 UAVs	3.7345E4 \pm 9.1964E3	3.1604E4 \pm 9.5034E3	2.9895E4 \pm 5.2438E3	2.2498E4 \pm 5.7047E3	3.2507E4 \pm 8.7148E3 \ddagger	14.88%
	3 UAVs	5.7976E4 \pm 9.5675E3	5.7438E4 \pm 1.2518E4 \ddagger	5.3212E4 \pm 1.1223E4	5.7064E4 \pm 9.5678E3	5.4463E4 \pm 9.4785E3	0.94%
	4 UAVs	7.8199E4 \pm 2.3309E4	6.7178E4 \pm 2.4483E4	6.8864E4 \pm 1.1419E4	4.4593E4 \pm 8.3818E3	7.9433E4 \pm 1.8065E4 \ddagger	-1.55%
Map3	2 UAVs	3.4258E4 \pm 1.0273E4	3.1642E4 \pm 7.5068E3 \ddagger	3.0620E4 \pm 6.6310E3	3.0512E4 \pm 4.6841E3	2.4705E4 \pm 5.3395E3	8.27%
	3 UAVs	6.6564E4 \pm 9.4940E3	4.1049E4 \pm 1.0347E4	5.3744E4 \pm 1.7804E4	5.5807E4 \pm 6.7582E3 \ddagger	4.5691E4 \pm 7.4382E3	19.28%
	4 UAVs	6.5987E4 \pm 1.1272E4	5.9737E4 \pm 1.3179E4	5.9155E4 \pm 6.7712E3	6.6650E4 \pm 1.8444E4 \ddagger	6.3182E4 \pm 1.1302E4	-0.99%
Map4	2 UAVs	3.2183E4 \pm 8.6605E3	2.4435E4 \pm 5.4714E3	2.9131E4 \pm 1.0785E4	2.5815E4 \pm 3.5943E3	2.9847E4 \pm 9.1883E3 \ddagger	7.83%
	3 UAVs	5.2952E4 \pm 1.1475E4	4.5824E4 \pm 7.5349E3	4.7387E4 \pm 8.9396E3	4.8111E4 \pm 7.7270E3	4.9192E4 \pm 1.2385E4 \ddagger	7.64%
	4 UAVs	8.8826E4 \pm 1.2021E4	7.5891E4 \pm 1.7997E4	8.8036E4 \pm 1.2731E4 \ddagger	8.7851E4 \pm 9.2991E3	7.3634E4 \pm 5.1694E3	1.11%
Map5	2 UAVs	4.2367E4 \pm 4.4300E3	2.9697E4 \pm 5.3265E3	2.9078E4 \pm 6.0260E3	2.9074E4 \pm 3.3913E3	3.0324E4 \pm 5.1485E3 \ddagger	39.71%
	3 UAVs	2.5200E4 \pm 5.0638E3	2.1708E4 \pm 5.5561E3	1.8945E4 \pm 2.1251E3	1.7166E4 \pm 5.6160E3	2.5826E4 \pm 6.8500E3 \ddagger	-2.42%
	4 UAVs	9.8716E4 \pm 5.3446E4	5.8384E4 \pm 7.8978E3	5.7040E4 \pm 1.0816E4	7.6483E4 \pm 1.6339E4 \ddagger	3.7036E4 \pm 5.3363E3	29.07%
Average	/	/	/	/	/	/	17.12%
-/+		15/0/0	15/0/0	14/1/0	13/2/0		

guidance in operator selection, which limits individuals' ability to make contextually relevant decisions. In terms of EP, EMMOP achieves improvements of over 1.77% in HV and 17.12% in PD, reflecting the combined effects of all strategies. Furthermore, Fig. 21 presents the results of the Friedman rank test for all EMMOP variants, indicating that EMMOP attains the best average rank for both HV and PD metrics.

Additionally, Fig. 22 illustrates the operator selection ratios for the main task of EMMOP during optimization in two different scenarios: Map 1 (3 UAVs) and Map 4 (3 UAVs). In Map 1, EMMOP primarily employs operator $U_{l,g}^3$ (labeled OP1) and operator $U_{l,g}^1$ (labeled OP4); in Map 4, operator $U_{l,g}^5$ (labeled OP3) and operator $U_{l,g}^2$ (labeled OP5) are

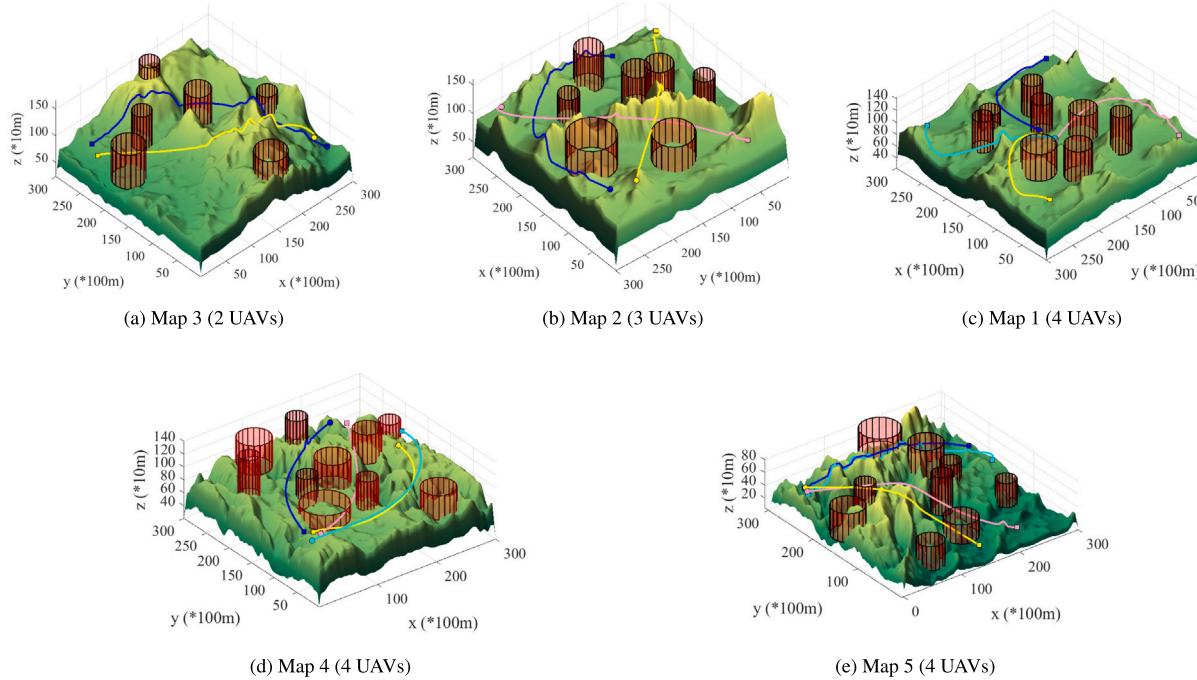


Fig. 18. 3D perspective of optimal compromise paths produced by EMMOP for five test cases.

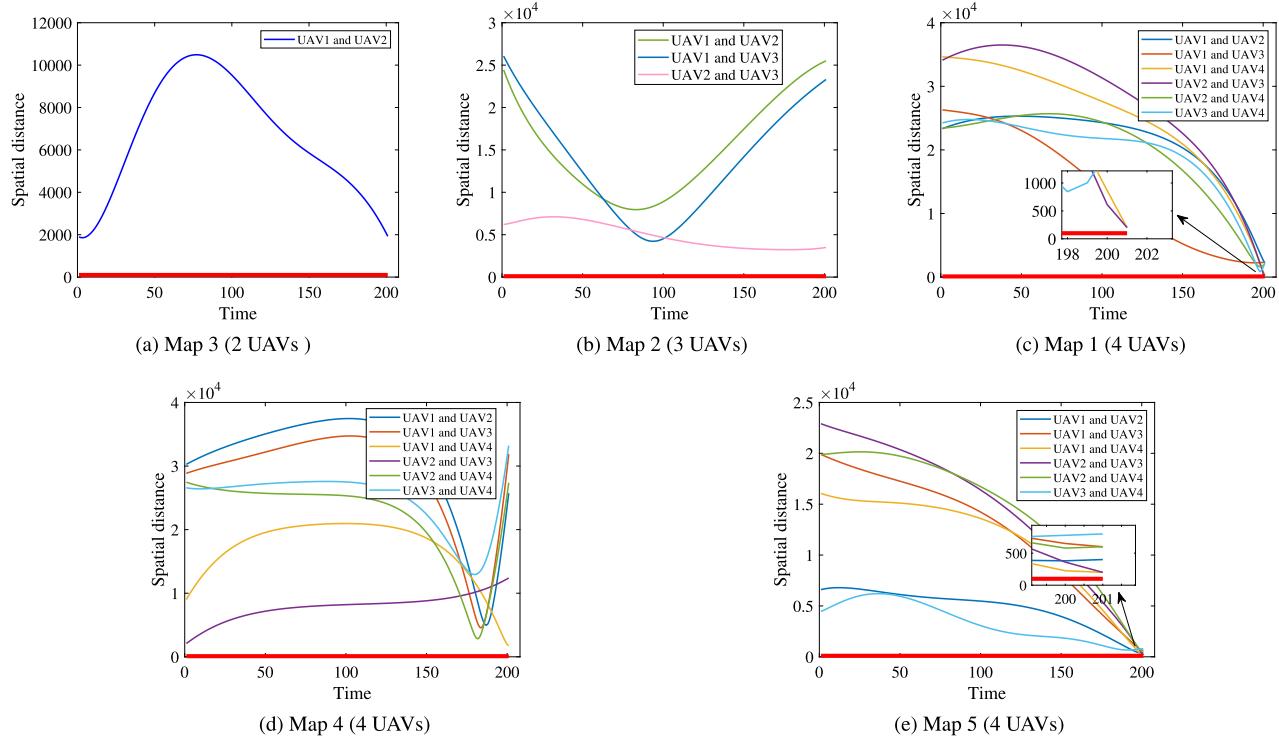


Fig. 19. Distance between each pair of UAVs in five test cases.

predominantly selected. This behavior indicates that EMMOP adapts its operator combination to the specific characteristics of each problem instance, thereby supporting the effectiveness of the DAOS strategy. Overall, EMMOP demonstrates stronger performance and competitiveness than all its variants, validating the positive effects of the proposed improvements on the algorithm.

5.6. Validation of EMMOP performance using V-REP simulation

To further assess the performance of EMMOP, we employ the V-REP simulator to model UAV flights along paths generated by the algorithm. V-REP is a versatile and scalable platform that integrates realistic

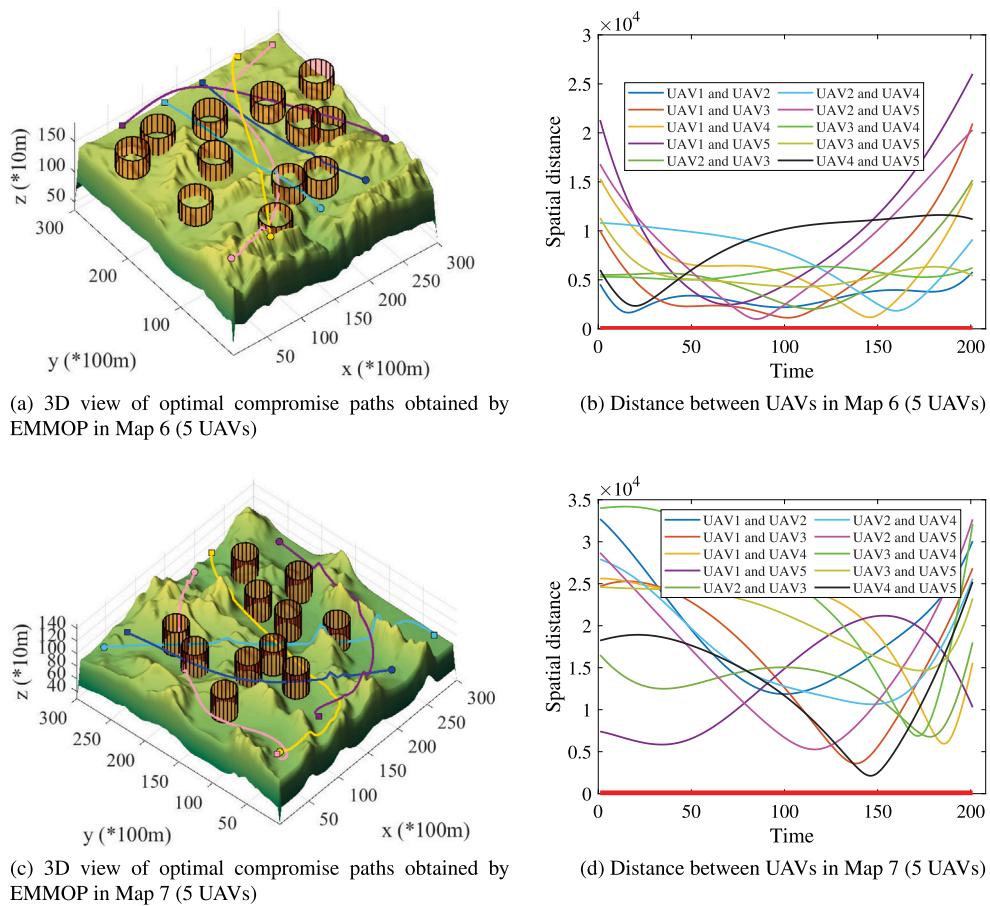


Fig. 20. Best compromise solutions obtained by EMMOP in Map 6 and Map 7.

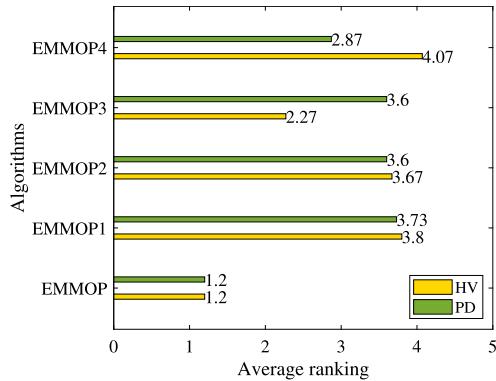


Fig. 21. Average rankings of EMMOP and its degradative variants based on the Friedman test.

UAV models and environmental obstacles, closely approximating real-world scenarios [72]. The simulated environment features mountains (displayed in red) and various threat types (shown in yellow). The UAV has a maximum flight speed of 50 m/s, a maximum turning angle of 60°, and a climbing slope limit of 45°. Fig. 23 illustrates the simulation process. In Fig. 23(a), the UAV's trajectory from the starting point (indicated by a purple dot) to the target point (indicated by a green dot) is shown along the path generated by EMMOP. Figs. 23(b) and 23(c) depict the UAV's threat avoidance maneuvers from two distinct perspectives, as captured by camera 1 and camera 2, respectively. These results confirm that EMMOP is capable of planning feasible, collision-free paths in a realistic simulated environment.

6. Conclusions and future works

In this study, we propose the EMMOP algorithm to address the MOMUPP problem, which aims to optimize both path length and risk cost simultaneously. Unlike existing methods, EMMOP introduces two key innovations. First, it employs DAOS that automatically allocates the most suitable search operators to two interrelated tasks within an advanced EMT framework, enhancing the generation of high-quality offspring. Second, it integrates a KTDF, which facilitates effective information exchange between these two tasks and improves overall performance. The performance of EMMOP is validated across 15 test cases, with experimental results highlighting its advantages in terms of convergence, diversity, robustness, and scalability compared to existing algorithms. Moreover, the effectiveness of the EMT framework, KTDF, and DAOS strategies has been confirmed.

Regarding future work, we plan to incorporate additional auxiliary tasks to further enhance the complementary evolutionary process and improve population diversity through more effective resource allocation [73]. Moreover, we intend to introduce dynamic threat modeling into the MOMUPP problem and address it using a rolling time domain approach [74]. However, each update of the rolling window may require executing the EMMOP algorithm one or more times, potentially resulting in significant computational overhead. To address this issue, we aim to develop a GPU [75]-based implementation of EMMOP that leverages hardware-level parallelism and high memory bandwidth to alleviate the computational burden.

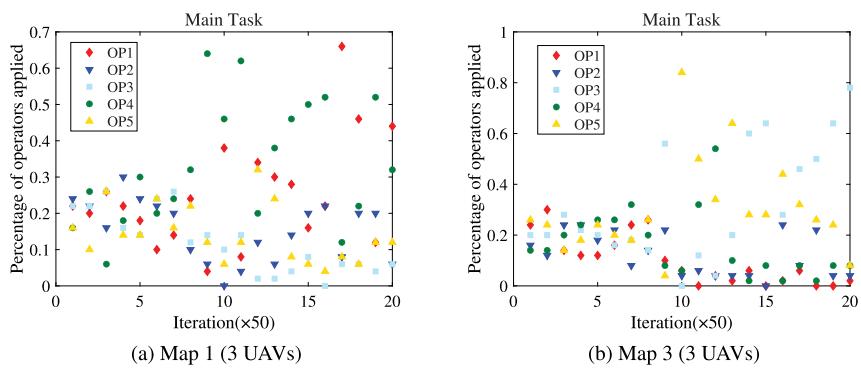


Fig. 22. Ratio of operators selected by EMMOP during the optimization process of solving two test cases.

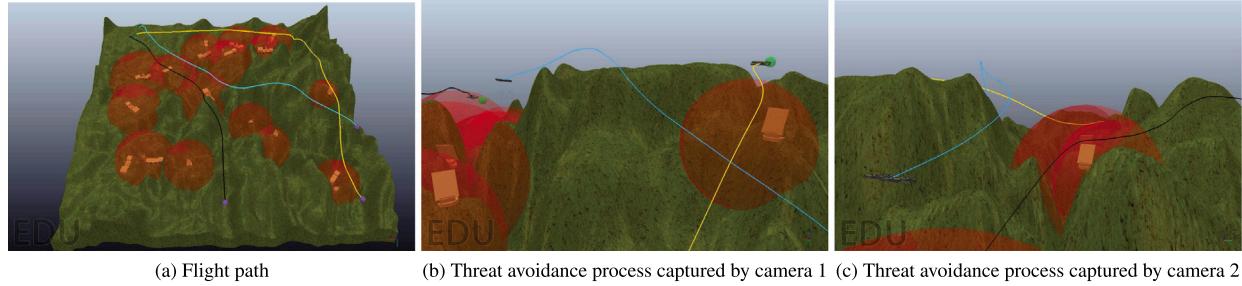


Fig. 23. Photos showing the simulation process of three UAVs in V-REP.

CRediT authorship contribution statement

Kai Meng: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Data curation, Conceptualization. **Binghong Wu:** Software, Project administration, Conceptualization. **Bin Xin:** Writing – review & editing, Supervision, Formal analysis, Conceptualization. **Fang Deng:** Funding acquisition, Formal analysis. **Chen Chen:** Supervision, Methodology, Funding acquisition.

Declaration of competing 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

This work is supported by the National Natural Science Foundation of China (Grant Nos. 62273044, 62022015, and 62088101).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.swevo.2025.102145>.

Data availability

The data that has been used is confidential.

References

- [1] M.F. Fan, H. Liu, G.H. Wu, A. Gunawan, G. Sartoretti, Multi-UAV reconnaissance mission planning via deep reinforcement learning with simulated annealing, *Swarm Evol. Comput.* 93 (2025) 101858.
- [2] Z.Z. Bai, H.Y. Zhou, J.M. Shi, L.N. Xing, J.Q. Wang, A hybrid multi-objective evolutionary algorithm with high solving efficiency for UAV defense programming, *Swarm Evol. Comput.* 87 (2024) 101572.
- [3] Z. Zhang, C. Chen, L.D. Wang, Y.L. Ding, F. Deng, A two-phase planner for messenger routing problem in UAV-UGV coordination systems, *IEEE Trans. Autom. Sci. Eng.* 22 (2025) 16948–16963.
- [4] B. Yıldız, M.F. Aslan, A. Durdu, A. Kayabasi, Consensus-based virtual leader tracking swarm algorithm with GDRRT*-PSO for path-planning of multiple-UAVs, *Swarm Evol. Comput.* 88 (2024) 101612.
- [5] X.Z. Chai, Z.S. Zheng, J.M. Xiao, L. Yan, B.Y. Qu, P.W. Wen, H.Y. Wang, Y. Zhou, H. Sun, Multi-strategy fusion differential evolution algorithm for UAV path planning in complex environment, *Aerosp. Sci. Technol.* 121 (2022) 107287.
- [6] M. Jones, S. Djahel, K. Welsh, Path-planning for unmanned aerial vehicles with environment complexity considerations: A survey, *ACM Comput. Surv.* 55 (11) (2023) 1–39.
- [7] M.Y. Wang, D. Zhang, B.H. Wang, L. Li, Dynamic trajectory planning for multi-UAV multi-mission operations using a hybrid strategy, *IEEE Trans. Aerosp. Electron. Syst.* (2025).
- [8] S.J. Fusic, R. Sitharthan, Improved RRT* algorithm-based path planning for unmanned aerial vehicle in a 3D metropolitan environment, *Unmanned Syst.* 12 (05) (2024) 859–875.
- [9] H.L. Sheng, J. Zhang, Z.Y. Yan, B.X. Yin, S.Y. Liu, T.T. Bai, D.B. Wang, New multi-UAV formation keeping method based on improved artificial potential field, *Chin. J. Aeronaut.* 36 (11) (2023) 249–270.
- [10] K. Meng, C. Chen, T.Y. Wu, B. Xin, M.M. Liang, F. Deng, Evolutionary state estimation-based multi-strategy jellyfish search algorithm for multi-UAV cooperative path planning, *IEEE Trans. Intell. Veh.* (2024).
- [11] W.H. Zhang, C.D. Peng, Y. Yuan, J.R. Cui, L. Qi, A novel multi-objective evolutionary algorithm with a two-fold constraint-handling mechanism for multiple UAV path planning, *Expert Syst. Appl.* 238 (2024) 121862.
- [12] X.Y. Zhang, S. Xia, X.Z. Li, T. Zhang, Multi-objective particle swarm optimization with multi-mode collaboration based on reinforcement learning for path planning of unmanned air vehicles, *Knowl.-Based Syst.* 250 (2022) 109075.
- [13] H. Bai, T. Fan, Y. Niu, Z. Cui, Multi-UAV cooperative trajectory planning based on many-objective evolutionary algorithm, *Complex Syst. Model. Simul.* 2 (2) (2022) 130–141.
- [14] T.W. Zhou, Z.H. Zhou, H.Y. Qiu, B. Niu, G.X.-G. Yue, W. Pedrycz, Two-stage knowledge-assisted coevolutionary NSGA-II for bi-objective path planning of multiple unmanned aerial vehicles, *Swarm Evol. Comput.* 90 (2024) 101680.
- [15] K.Y. Zhong, F. Xiao, X.P. Gao, Three-dimensional dynamic collaborative path planning for multiple UCAVs using an improved NSGAII, *Clust. Comput.* 28 (2) (2025) 75.
- [16] J. Liang, X.X. Ban, K.J. Yu, B.Y. Qu, K.J. Qiao, C.T. Yue, K. Chen, K.C. Tan, A survey on evolutionary constrained multiobjective optimization, *IEEE Trans. Evol. Comput.* 27 (2) (2023) 201–221.
- [17] J. Xu, C.Y. Xie, L.R. Ma, L. Yang, T. Zhang, Multi-objective evolutionary algorithm with two balancing mechanisms for heterogeneous UAV swarm path planning, *Appl. Soft. Comput.* (2025) 112927.

- [18] Y.T. Wan, Y.F. Zhong, A.L. Ma, L.P. Zhang, An accurate UAV 3-D path planning method for disaster emergency response based on an improved multiobjective swarm intelligence algorithm, *IEEE Trans. Cybern.* 53 (4) (2023) 2658–2671.
- [19] J.H. Zhu, X.B. Yu, F. Wang, Y.Q. Mao, Multi-objective optimal power flow problem using constrained dynamic multitasking multi-objective optimization algorithm, *Swarm Evol. Comput.* 93 (2025) 101850.
- [20] K.J. Qiao, K.J. Yu, B.Y. Qu, J. Liang, H. Song, C.T. Yue, An evolutionary multitasking optimization framework for constrained multiobjective optimization problems, *IEEE Trans. Evol. Comput.* 26 (2) (2022) 263–277.
- [21] R. Li, L. Wang, W.Y. Gong, F. Ming, An evolutionary multitasking memetic algorithm for multi-objective distributed heterogeneous welding flow shop scheduling, *IEEE Trans. Evol. Comput.* (2024).
- [22] Y. Jiang, X.X. Xu, M.Y. Zheng, Z.H. Zhan, Evolutionary computation for unmanned aerial vehicle path planning: a survey, *Artif. Intell. Rev.* 57 (10) (2024) 267.
- [23] M.H. Zhang, H.L. Wang, Z.Y. Li, Y.X. Wang, X.L. Zhang, Q. Tang, S.C. Ma, J.F. Wu, Fluid-based moderate collision avoidance for UAV formation in 3-D low-altitude environments, *Chin. J. Aeronaut.* (2024).
- [24] Y. Liu, X.J. Zhang, Y. Zhang, X.M. Guan, Collision free 4D path planning for multiple UAVs based on spatial refined voting mechanism and PSO approach, *Chin. J. Aeronaut.* 32 (6) (2019) 1504–1519.
- [25] S.K. Shao, Y. Peng, C.L. He, Y. Du, Efficient path planning for UAV formation via comprehensively improved particle swarm optimization, *ISA Trans.* 97 (2020) 415–430.
- [26] W.J. He, X.G. Qi, L.F. Liu, A novel hybrid particle swarm optimization for multi-UAV cooperative path planning, *Appl. Intell.* 51 (10) (2021) 7350–7364.
- [27] L. Xu, X.B. Cao, W.B. Du, Y.M. Li, Cooperative path planning optimization for multiple UAVs with communication constraints, *Knowl.-Based Syst.* 260 (2023) 110164.
- [28] Y.B. Niu, X.F. Yan, Y.Z. Wang, Y.Z. Niu, Three-dimensional collaborative path planning for multiple UCAVs based on improved artificial ecosystem optimizer and reinforcement learning, *Knowl.-Based Syst.* 276 (2023) 110782.
- [29] T.S. Lou, Y. Wang, Z.P. Yue, L.Y. Zhao, Multi-UAV collaborative trajectory planning for 3D terrain based on CS-GJO algorithm, *Complex Syst. Model. Simul.* 4 (3) (2024) 274–291.
- [30] A. Gupta, Y.-S. Ong, L. Feng, Multifactorial evolution: Toward evolutionary multitasking, *IEEE Trans. Evol. Comput.* 20 (3) (2015) 343–357.
- [31] A. Gupta, Y.-S. Ong, L. Feng, K.C. Tan, Multiobjective multifactorial optimization in evolutionary multitasking, *IEEE Trans. Cybern.* 47 (7) (2016) 1652–1665.
- [32] S.Q. Yang, Y.T. Qi, R. Yang, X.L. Ma, H.B. Zhang, A surrogate assisted evolutionary multitasking optimization algorithm, *Appl. Soft. Comput.* 132 (2023) 109775.
- [33] X.L. Wu, W. Wang, H.Y. Yang, H.G. Han, J.F. Qiao, Multitasking feedback optimization algorithm based on an evolutionary state estimator, *IEEE Trans. Emerg. Top. Comput. Intell.* 8 (3) (2024) 2554–2569.
- [34] Y.C. Li, D.C. Li, W.Y. Gong, Q. Gu, Multiobjective multitask optimization via diversity- and convergence-oriented knowledge transfer, *IEEE Trans. Syst. Man Cybern.: Syst.* 55 (3) (2025) 2367–2379.
- [35] M. Li, B. Liu, B. Xin, L. Feng, P. Li, Accelerate solving expensive scheduling by leveraging economical auxiliary tasks, 2024, arXiv preprint [arXiv:2404.01018](https://arxiv.org/abs/2404.01018).
- [36] Q.Z. Lin, Z.J. Wu, L.J. Ma, M.G. Gong, J.Q. Li, C.A.C. Coello, Multiobjective multitasking optimization with decomposition-based transfer selection, *IEEE Trans. Cybern.* 54 (5) (2024) 3146–3159.
- [37] Q.X. Shang, L.J. Zhang, L. Feng, Y.Q. Hou, J.H. Zhong, A. Gupta, K.C. Tan, H.L. Liu, A preliminary study of adaptive task selection in explicit evolutionary many-tasking, in: 2019 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2019, pp. 2153–2159.
- [38] D.Z. Zhang, K.J. Yu, J. Liang, K.J. Qiao, B.Y. Qu, K. Chen, C.T. Yue, L. Wang, History-assisted two-state auxiliary task collaboration approach for dynamic constrained multiobjective optimization, *IEEE Trans. Evol. Comput.* (2024) 1–1.
- [39] J. Yi, J.R. Bai, H.B. He, W. Zhou, L.Z. Yao, A multifactorial evolutionary algorithm for multitasking under interval uncertainties, *IEEE Trans. Evol. Comput.* 24 (5) (2020) 908–922.
- [40] H.D. Wang, L. Feng, Y.C. Jin, J. Doherty, Surrogate-assisted evolutionary multitasking for expensive minimax optimization in multiple scenarios, *IEEE Comput. Intell. Mag.* 16 (1) (2021) 34–48.
- [41] S.X. Ding, T. Zhang, C. Chen, Y.S. Lv, B. Xin, Z.M. Yuan, R.S. Wang, P.M. Pardalos, An efficient particle swarm optimization with evolutionary multitasking for stochastic area coverage of heterogeneous sensors, *Inf. Sci.* 645 (2023) 119319.
- [42] K. Chen, B. Xue, M.J. Zhang, F.Y. Zhou, Evolutionary multitasking for feature selection in high-dimensional classification via particle swarm optimization, *IEEE Trans. Evol. Comput.* 26 (3) (2022) 446–460.
- [43] C. Yang, Q.J. Chen, Z.X. Zhu, Z.-A. Huang, S.L. Lan, L.H. Zhu, Evolutionary multitasking for costly task offloading in mobile-edge computing networks, *IEEE Trans. Evol. Comput.* 28 (2) (2024) 338–352.
- [44] Y. Hou, Y.J. Shen, H.G. Han, J.J. Wang, Multi-task differential evolution algorithm with dynamic resource allocation: A study on e-waste recycling vehicle routing problem, *Swarm Evol. Comput.* 92 (2025) 101806.
- [45] C.E. Yang, J.L. Ding, Y.C. Jin, C.Z. Wang, T.Y. Chai, Multitasking multiobjective evolutionary operational indices optimization of beneficiation processes, *IEEE Trans. Autom. Sci. Eng.* 16 (3) (2018) 1046–1057.
- [46] J.Y. Pei, Y. Mei, J.L. Liu, M.J. Zhang, X. Yao, Adaptive operator selection for meta-heuristics: A survey, *IEEE Trans. Artif. Intell.* (2025).
- [47] Y.J. Song, Y.T. Wu, Y.Y. Guo, R. Yan, P.N. Suganthan, Y. Zhang, W. Pedrycz, S. Das, R. Mallipeddi, O.S. Ajani, et al., Reinforcement learning-assisted evolutionary algorithm: A survey and research opportunities, *Swarm Evol. Comput.* 86 (2024) 101517.
- [48] Z.J. Lin, K.Z. Gao, P.Y. Duan, N.Q. Wu, P.N. Suganthan, Prediction and feedback assisted evolutionary algorithms for scheduling urban traffic signals, *IEEE Trans. Intell. Transp. Syst.* (2025) 1–10.
- [49] R. Li, W.Y. Gong, C. Lu, A reinforcement learning based RMOEA/D for bi-objective fuzzy flexible job shop scheduling, *Expert Syst. Appl.* 203 (2022) 117380.
- [50] S. Yin, Z. Xiang, Adaptive operator selection with dueling deep Q-network for evolutionary multi-objective optimization, *Neurocomputing* 581 (2024) 127491.
- [51] R. Li, W.Y. Gong, L. Wang, C. Lu, Z.X. Pan, X.Y. Zhuang, Double DQN-based coevolution for green distributed heterogeneous hybrid flowshop scheduling with multiple priorities of jobs, *IEEE Trans. Autom. Sci. Eng.* 21 (4) (2023) 6550–6562.
- [52] C.Z. Qu, W.D. Gai, M.Y. Zhong, J. Zhang, A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning, *Appl. Soft Comput.* 89 (2020) 106099.
- [53] X.B. Yu, W.G. Luo, Reinforcement learning-based multi-strategy cuckoo search algorithm for 3D UAV path planning, *Expert Syst. Appl.* 223 (2023) 119910.
- [54] W.T. Wang, X.L. Li, J. Tian, UAV formation path planning for mountainous forest terrain utilizing an artificial rabbit optimizer incorporating reinforcement learning and thermal conduction search strategies, *Adv. Eng. Inf.* 62 (2024) 102947.
- [55] H.W. Zhan, Y. Zhang, J.B. Huang, Y.J. Song, L.N. Xing, J. Wu, Z.Y. Gao, A reinforcement learning-based evolutionary algorithm for the unmanned aerial vehicles maritime search and rescue path planning problem considering multiple rescue centers, *Memet. Comput.* 16 (3) (2024) 373–386.
- [56] Z.Q. Zhang, F.C. Wu, B. Qian, R. Hu, L. Wang, H.P. Jin, A Q-learning-based hyper-heuristic evolutionary algorithm for the distributed flexible job-shop scheduling problem with crane transportation, *Expert Syst. Appl.* 234 (2023) 121050.
- [57] Y.J. Song, J.W. Ou, W. Pedrycz, P.N. Suganthan, X.W. Wang, L.N. Xing, Y. Zhang, Generalized model and deep reinforcement learning-based evolutionary method for multitype satellite observation scheduling, *IEEE Trans. Syst. Man Cybern.: Syst.* 54 (4) (2024) 2576–2589.
- [58] C. Wang, M.M. Cao, H. Jiang, X.S. Xiang, X.Y. Zhang, A deep reinforcement learning-based adaptive large neighborhood search for capacitated electric vehicle routing problems, *IEEE Trans. Emerg. Top. Comput. Intell.* 9 (1) (2025) 131–144.
- [59] Y.J. Zheng, X.C. Xie, Z.Y. Zhang, J.T. Shi, Deep reinforcement learning assisted memetic scheduling of drones for railway catenary deicing, *Swarm Evol. Comput.* 91 (2024) 101719.
- [60] S. Aslan, An immune plasma algorithm with a modified treatment schema for UCAV path planning, *Eng. Appl. Artif. Intell.* 112 (2022) 104789.
- [61] P. Yang, K. Tang, J.A. Lozano, X.B. Cao, Path planning for single unmanned aerial vehicle by separately evolving waypoints, *IEEE Trans. Robot.* 31 (5) (2015) 1130–1146.
- [62] K.R. Opara, J. Arabas, Differential evolution: A survey of theoretical analyses, *Swarm Evol. Comput.* 44 (2019) 546–558.
- [63] L.Q. Pan, W.T. Xu, L.H. Li, C. He, R. Cheng, Adaptive simulated binary crossover for rotated multi-objective optimization, *Swarm Evol. Comput.* 60 (2021) 100759.
- [64] X.Y. Zhang, X.T. Zheng, R. Cheng, J.F. Qiu, Y.C. Jin, A competitive mechanism based multi-objective particle swarm optimizer with fast convergence, *Inf. Sci.* 427 (2018) 63–76.
- [65] K.J. Qiao, J. Liang, K.J. Yu, C.T. Yue, H.Y. Lin, D.Z. Zhang, B.Y. Qu, Evolutionary constrained multiobjective optimization: Scalable high-dimensional constraint benchmarks and algorithm, *IEEE Trans. Evol. Comput.* 28 (4) (2024) 965–979.
- [66] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm, *TIK Rep.* 103 (2001).
- [67] S.B. Liu, Z.Y. Wang, Q.Z. Lin, J.Q. Li, K.C. Tan, Learning-aided evolutionary search and selection for scaling-up constrained multiobjective optimization, *IEEE Trans. Evol. Comput.* (2024) 1–1.
- [68] F. Ming, W.Y. Gong, L. Wang, Y.C. Jin, Constrained multi-objective optimization with deep reinforcement learning assisted operator selection, *IEEE/CAA J. Autom. Sin.* 11 (4) (2024) 919–931.
- [69] F. Ye, P. Duan, L.L. Meng, H.Y. Sang, K.Z. Gao, An enhanced artificial bee colony algorithm with self-learning optimization mechanism for multi-objective path planning problem, *Eng. Appl. Artif. Intell.* 149 (2025) 110444.

- [70] P. Duan, Z.A. Yu, K.Z. Gao, L.L. Meng, Y.Y. Han, F. Ye, Solving the multi-objective path planning problem for mobile robot using an improved NSGA-II algorithm, *Swarm Evol. Comput.* 87 (2024) 101576.
- [71] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm Evol. Comput.* 1 (1) (2011) 3–18.
- [72] E. Rohmer, S.P. Singh, M. Freese, V-REP: A versatile and scalable robot simulation framework, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2013, pp. 1321–1326.
- [73] K.J. Qiao, J. Liang, K.J. Yu, X.X. Ban, C.T. Yue, B.Y. Qu, P.N. Suganthan, Constraints separation based evolutionary multitasking for constrained multi-objective optimization problems, *IEEE/CAA J. Autom. Sin.* 11 (8) (2024) 1819–1835.
- [74] H.W. Li, Q.Z. Luo, Y. Gu, C.Z. Fan, G.H. Wu, Multi-objective dynamic scheduling optimization method for relay satellites based on rolling horizon strategy, *Acta Astronaut. Sin.* 45 (16) (2024).
- [75] F.T. Li, Y. Yin, X. Wang, C.Y. Yu, Y.H. Zhao, Y. Tian, GPU-powered evolutionary auxiliary multitasking for fast SNP interaction detection, *IEEE Trans. Comput. Biol. Bioinf.* (2025) 1–12.