Swarm and Evolutionary Computation 90 (2024) 101680

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Two-stage knowledge-assisted coevolutionary NSGA-II for bi-objective path planning of multiple unmanned aerial vehicles

Tianwei Zhou [a], Zhenghan Zhou [a], Haiyun Qiu [a], Ben Niu [a,b,*], Gabriel Xiao-Guang Yue [c], Witold Pedrycz [d,e,f]

[a] College of Management, Shenzhen University, Shenzhen, China
[b] School of Computer Science, Shenzhen Institute of Information Technology, Shenzhen, China
[c] Department of Computer Science and Engineering, European University Cyprus, Nicosia, Cyprus
[d] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada
[e] Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah, Saudi Arabia
[f] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

ARTICLE INFO

Keywords:
Unmanned aerial vehicles
Path planning
Energy consumption
Coevolution
Reinforcement learning

ABSTRACT

This paper focuses on the bi-objective path planning problem of multiple unmanned aerial vehicles (UAVs) under the complex environment with numerous obstacles and threat areas, where the UAVs need to be kept as far away as possible from threat areas during flight. Based on the integrated energy reduction perspective, a bi-objective model is subtly constructed by minimizing the total energy consumption of each path (including flight altitude, horizontal turns, and path length), and minimizing the costs of the total threats (including ground radar, anti-aircraft gun, missile and geological hazard threat areas). Moreover, a two-stage knowledge-assisted coevolutionary NSGA-II algorithm is novelly proposed to enhance collaboration and avoid collision. The first stage is designed for population convergence, where the considered constrained problem is solved with the help of the designed problem without the constraints of threats and obstacles. The second stage emphasizes the quality and diversity of solutions. In this stage, a double-population coevolution approach is developed. Additionally, a multi-mode strategy is introduced for the inferior population, leveraging reinforcement learning. This strategy aids in selecting the optimal mode from random swing, directed guidance, and potential dominance exploration. Furthermore, experimental results in two different environments show that the proposed algorithm can better solve the collaborative path planning problem for multiple UAVs compared with other five classical or recent proposed algorithms.

1. Introduction

Unmanned aerial vehicle (UAV), a semi-autonomous aircraft without onboard pilot [1], has become widely used nowadays in both civilian and military [2,3], such as rescue [4], disaster monitoring [5], wildfire tracking [6], and so on. However, in practical applications, single UAV is hard to deal with complex missions, such as the one with dispersed task points, long-time consumption, or inconsistent demands at each task point [7]. UAV group, with better robustness and lower cost, is often utilized to solve this problem by cooperation [8,9]. In UAV group cooperation, path planning is one of the most critical problem [10,11]. The quality of the generated path for each UAV is affected by both safety and energy cost, so how to find a safe and reasonable shortest path from the starting position to the destination position to avoid potential threats becomes our main concern.

Presently, the studies about path planning problem of multiple UAVs under complex environment with numerous obstacles and threat areas have attracted the attentions of the scholars from both academia and industry. In existing studies, both single-objective [12] and multi-objective [13] models are constructed. However, few studies provide scheduling scheme based on the energy perspective. Moreover, most works are related to the discussion on the improvement on multi-objective evolutionary algorithm [14] or Q-learning [15], while few

* Corresponding author at: College of Management, Shenzhen University, Shenzhen, China.
E-mail address: drniuben@gmail.com (B. Niu).

https://doi.org/10.1016/j.swevo.2024.101680
Received 29 April 2024; Received in revised form 23 June 2024; Accepted 21 July 2024
Available online 27 July 2024

reports are related to the tailored combination of the above two algorithms based on problem characteristics, which leaves a huge room for algorithm improvement.

Based on the above analyses, in perspective of energy consumption, optimizing the flight path of UAVs, and using the adaptive and autonomous characteristics of the Q-learning algorithm to enhance the search efficiency of the algorithm to solve the UAV path planning problem has become our focus. To address this problem, a unique bi-objective model is constructed by merging the UAV path properties in the energy consumption view, and the two-stage knowledge-assisted coevolutionary NSGA-II (TSKAC-NSGA-II) algorithm is introduced to more effectively generate paths for multiple UAVs. The originality of this paper comes from below three folds, which are proposed based on the embedding of the problem characteristics.

(1) A bi-objective path planning model for multiple UAVs is proposed with a focus on energy consumption and multiple UAVs cooperation. This model takes into account the energy cost associated with path length, flight height, and turning angles in a comprehensive manner.
(2) The two-stage knowledge-assisted coevolutionary NSGA-II algorithm is novelly proposed to guarantee both convergence and diversity. For the first stage, an assistant population for the corresponding designed problem without constraints of threats and obstacles, is utilized to help the main population with the constraint problem. While for the second stage, the framework of Q-learning is embedded in the multi-mode strategy in the double-population coevolution approach to improve population diversity.
(3) In multi-mode strategy, three modes are developed, including random swing mode focusing on randomness, directed guided mode focusing on guidance of dominant subpopulation's mean value, and potential dominance exploration mode focusing on exploration around the dominant subpopulation. Moreover, two metrics, including relative distance and set optimality, are constructed to categorize the inferior subpopulation into four states based on the distribution of the number of particles and the relative distance relationship with the superior offspring population.

The following part of this paper is organized as follows. Section 2 is a literature review. Section 3 establishes the bi-objective UAV path planning model for multiple UAVs cooperation. Section 4 provides a detailed introduction to the proposed TSKACNSGA-II algorithm. In Section 5, two different environments are considered in our simulations and the superiority of our proposed algorithm is verified though the comparisons against other five algorithms. Finally, Section 6 presents the conclusion and future direction.

2. Literature review

To deal with path planning problem, different scholars have developed various models [16]. For single-objective optimization, Yu et al. constructed the model with the objective of obtaining the shortest path and the constraints, including turning angles and threat levels [12]. In view of the fact that the performance of the UAV flight path is related to both cost and flight safety, Zhang et al. proposed a bi-objective model aimed at minimizing both distance and altitude costs while taking into account constraints related to path safety., such as dangerous areas, turning angles, terrain obstacles, no-fly zones, climbing, and gliding slopes [13]. In addressing the collaborative multiple UAVs problem, they included collision avoidance between UAVs as part of the second objective. Alternative paths for collision detection were derived, resulting in a constrained set of options. In these studies, some related factors, such as flight path length, climbing and gliding slope, turning angles and flight height, are separately considered. However, in mechanical view of energy consumption, these factors are in fact mutually interconnected [17]. Therefore, this study integrates these factors into energy consumption-related factors and sets energy cost minimization as the first objective, ensuring that the generated trajectory saves as much energy as possible. In addition, most current UAV path planning models focus on single-objective optimization [12,18], but there are many factors that actually affect the quality of path, making it complex to assign weights to each factor. To overcome these limitations, this study has established a bi-objective model aimed at minimizing energy cost and safety cost for the path.

To solve the path planning models, various scholars have proposed a wide variety of methods. The first category is sampling based, including probability road map (PRM) [19,20], rapid-exploration random tree (RRT) [21], and its improved versions. In these algorithms, calculation efficiency can be guaranteed, even for problems with high dimensions. However, the paths obtained by these methods are only feasible rather than optimal, leading to potential improvement opportunities. The second category is graph oriented, including A* [22], Dijkstra [23], etc. In these algorithms, although best solution can be obtained, but both spatial partitioning, generation and storage of the graphs are time-consuming, especially in complex environment. The third category is mathematical programming, including dynamic programming (DP) [24], mixed integer linear programming (MILP) [25], etc. These methods can obtain the best solution for problems with small-scale. For the large-scale problem, the computational time is exponentially increased, which is unacceptable in practical applications.

To deal with large-scale path planning problems in complex environment, artificial intelligence methods [26], including reinforcement learning (RL) [27–32], swarm intelligence (SI) algorithms [33,34], and evolutionary computation (EC) [35–37], become suitable choices. For RL algorithm, Zhao et al. used deep Q-learning for UAV path planning [38]. After that, Zhao et al. proposed a variant of deep Q-learning called wire fitting neural network Q-learning, and combined it with the hill climbing algorithm to obtain smooth paths in experiments [39]. However, these methods hold high computational costs. For SI or EC algorithms, Zhang et al. applied Q-learning to the selection of particle position update patterns based on the particle swarm optimization (PSO) algorithm [13]. Chai et al. proposed the biased PSO algorithm to address constrained trajectory planning problems [40]. Xu et al. proposed the CL-DMSPSO algorithm, enabling particles to learn from different individuals in different dimensions [41]. Sathyan et al. combined fuzzy logic with GA to improve the accuracy of the planned paths [42]. Chai et al. employed an enhanced NSGA-III algorithm to solve the constructed multi-objective constrained trajectory optimization model [43]. Through literature review, it is observed that the algorithms for UAVs path planning problems mostly consist of single-stage approaches, with few employing two-stage algorithms. Moreover, in complex environments with numerous obstacles and threat areas, these algorithms struggle to converge quickly and find optimal solutions due to the abundance of constraints, resulting in a limited feasible domain.

In order to address the issue of balancing convergence and diversity, the algorithms designed for constrained bi-objective optimization problems (CMOPs) attract our attention [14]. Tian et al. proposed a coevolutionary framework for CMOPs. This framework includes main populations for solving the constrained original problem and assistant populations for solving problems with partial objectives and constraints. In scenarios with limited constraints, the feasible region is larger, allowing the assistant population to converge rapidly. By sharing the offspring information between the two populations, the assistant population can facilitate the swift convergence of the main population. So the designed framework can better solve the constraint bi-objective problems [44]. However, although this algorithm can find relative good solutions as alternative paths, it cannot guarantee that all alternative paths are meaningful in practice. Therefore, this paper has designed a two-stage algorithm. In the first stage, the co-evolutionary concept of the algorithm is adopted to enable the population to converge quickly

and in the second stage, the diversity of dominant individuals in the population is increased to ensure the practical feasibility of alternative UAV paths.

3. UAVs path planning model

3.1. Problem formulation

In this paper, the path planning problem for multiple UAVs is considered. During path planning process, two factors attract our attentions. One is energy consumption. Dai et al. [45] proposed a generalized propulsion energy consumption model (PECM) for rotary-wing UAVs that accounts for changes in the actual thrust-to-weight ratio (TWR) due to acceleration, velocity, and directional changes, and validated the model through experiments. This model demonstrates that both ascending and turning maneuvers, which alter the UAV's velocity, result in increased work and consequently, greater energy expenditure. Thibbotuwawa et al. [46] analyzed that ascending in a UAV requires additional thrust to overcome gravity and air resistance, leading to energy consumption. Based on Newton's laws of motion and aerodynamics, it can be inferred that: UAV ascent necessitates extra thrust to counteract gravity, while turning requires lateral forces and increased thrust. Even during stable flight, continuous work is needed to overcome air resistance, thereby incurring a sustained power requirement. Therefore, flight altitude, horizontal turns, and path length are significantly correlated with energy consumption. Moreover, energy consumption is not the sole criterion for measuring the quality of a UAV's flight path. If the path is unsafe, even the lowest energy expenditure is impractical. So another factor is safety guarantee. Hence, some threats, including radars, antiaircraft guns, missiles and geological hazard threats, need to be considered during path generation process. Additionally, for multiple UAVs cooperation, how to avoid mutual collision is another important problem.

Based on the above description, the considered problem holds bi-objective characteristics. With fixed starting positions, destination positions, as well as environmental factors and threats, the goal of this problem is to guide multiple UAVs to fly to different destination positions with the least energy consumption while avoiding collisions and threats. Moreover, for analysis convenience, the ground radar, anti-aircraft gun, missile and geological hazard threat areas are represented by cylinders. We need to ensure that the paths are as far away as possible from the threat areas.

Remark 1. Different from the existing literature, where costs are separately considered, we analyze the mutual relationship among different cost factors in the UAVs path planning process and merge them in an energy framework. Moreover, instead of treating safety as the constraints, it is better to build the model based on the bi-objective characteristics between energy costs and threat costs. When safety can be guaranteed, our main focus is to find the path with the least energy cost. Moreover, when energy costs are similar, the path with the furthest distance from threats is our better choice.

3.2. Path representation

In this study, we assume that the flight speeds of the UAVs are fixed, and the positions of threat areas are known. We set up $\tilde{L}$ path control points for each UAV and used cubic B-spline curves to smooth the path to meet the actual flight requirements [47]. The path of each UAV consists of the starting position S, the destination position D, and $N-2$ waypoints between S and D. The coordinates of the waypoint j of the UAV i are represented by $p_{i,j}=(x_{i,j},y_{i,j},z_{i,j})$, where $i=\{1,2,\dots,I\}$; $j=\{1,2,\dots,N-2\}$. Thus the complete path of the UAV i is $(p_{i,S},p_{i,1},p_{i,2},\dots,p_{i,N-2},p_{i,D})$. Control nodes are some special points used to measure the threat cost, which are evenly distributed and uniformly separate the path of the UAV into L parts.

3.3. Bi-objective model of UAVs path planning problem

In this subsection, we first present the objectives, including energy costs and threat costs. Then, the constraints for multiple UAVs are provided. In our model, threats are reflected by both objectives and constraints.

3.3.1. Bi-objective cost functions

Based on the above analysis, the objectives of the model are formulated by energy and threat costs. For each UAV i, the energy costs are described by length cost $C_{i,L}$, height cost $C_{i,H}$, turning angles cost $C_{i,T}$. And the threat costs are constructed by radar cost $P_{i,R}$, anti-aircraft guns cost $P_{i,G}$, missiles cost $P_{i,M}$, and geological hazard cost $P_{i,H}$. The bi-objective function is shown as follows:

$$\min f_i=(Obj_{i,1},Obj_{i,2}) \tag{1}$$

and

$$Obj_{i,1}=\beta_1\cdot C_{i,L}+\beta_2\cdot C_{i,H}+\beta_3\cdot C_{i,T}$$

$$Obj_{i,2}=\lambda_1\cdot P_{i,R}+\lambda_2\cdot P_{i,G}+\lambda_3\cdot P_{i,M}+\lambda_4\cdot P_{i,H}$$

where β_1, β_2, and β_3 are the weight coefficients related to the length cost, height cost and turning angles cost, respectively. And λ_1, λ_2, λ_3 and λ_4 are coefficients related to the radars, anti-aircraft guns, missiles and geological hazard threat costs, respectively. It is clear that the smaller $Obj_{i,1}$ is, the less energy the path will cost for UAV i. Moreover, for each path with no mutual collision or threats, the smaller $Obj_{i,2}$ is, the safer situation the path will face.

(1) *Length cost* $C_{i,L}$

With the fixed flight speed, the shorter the path distance is, the less energy the UAV will consume. The length cost $C_{i,L}$ is calculated by the comparison between path length and the distance from the starting position to the destination position, and is described as follows:

$$C_{i,L}=1-\frac{L_{i,SD}}{L_i} \tag{2}$$

with

$$L_{i,SD}=\sqrt{(x_{i,D}-x_{i,S})^2+(y_{i,D}-y_{i,S})^2+(z_{i,D}-z_{i,S})^2}$$

$$L_i=\sum_{j=1}^{N-1}\sqrt{(x_{i,j+1}-x_{i,j})^2+(y_{i,j+1}-y_{i,j})^2+(z_{i,j+1}-z_{i,j})^2}$$

where $L_{i,SD}$ represents the distance from the starting position to the destination position, L_i indicates the path length, $C_{i,L}$ varies between 0 and 1.

(2) *Height cost* $C_{i,H}$

Energy consumption can be decreased by reducing the frequency and magnitude of the ascents. In this paper, height cost $C_{i,H}$ is represented by the path's vertical stability and is described as follows:

$$C_{i,H}=\sqrt{\frac{\sum_{j=1}^{N-1}(\bar{h}_{i,j}-h_{i,m})^2}{N-1}} \tag{3}$$

with

$$\bar{h}_{i,j}=\frac{h_{i,j}-h_{i,S}}{h_{i,G}-h_{i,S}}$$

$$h_{i,m}=\frac{\sum_{j=1}^{N-1}\bar{h}_{i,j}}{N-1}$$

where $\bar{h}_{i,j}$ represents the standardized flight height of the waypoint j, and $h_{i,m}$ denotes the mean value of all standardized flight heights of the waypoints. $h_{i,G}$ is the highest flight height and $h_{i,S}$ is the lowest one among the waypoints. The value of $C_{i,H}$ varies between 0 and 1, which is smaller when the path is more stable and has fewer fluctuations.

(3) *Turning angles cost* $C_{i,T}$

Energy consumption can be decreased by reducing the frequency and magnitude of turns. In this paper, turning angles cost $C_{i,T}$ is represented by the path's horizontal stability, which is calculated by the comparison between horizontal angle and its mean value, and is described as follows:

$$C_{i,T} = \sqrt{\frac{\sum_{j=2}^{N-1}(\alpha_{i,j} - \alpha_{i,m})^2}{N-2}} \tag{4}$$

with

$$\alpha_{i,j} = \frac{\arccos \frac{(p_{i,j}-p_{i,j-1})\cdot(p_{i,j+1}-p_{i,j})}{\|p_{i,j}-p_{i,j-1}\|\cdot\|p_{i,j+1}-p_{i,j}\|}}{\alpha_{i,\max}}$$

$$\alpha_{i,m} = \frac{\sum_{j=2}^{N-1}\alpha_{i,j}}{N-2}$$

where $\alpha_{i,\max}$ is the maximum limit of the horizontal angle, and $\alpha_{i,j}$ is the normalized value of the horizontal angle j. $\alpha_{i,m}$ is the mean value of the normalized horizontal angles $\alpha_{i,j}$. The value of $C_{i,T}$ varies between 0 and 1.

Remark 2. As for the flight height and horizontal turning angle of the path, many researchers simply sum up the flight altitudes of each waypoint and the total horizontal turning angles of each path segment. However, this method struggles to comprehensively consider energy consumption. Therefore, this study transforms the measurement standards for altitude and horizontal turning angles into expressions related to mean square deviation and constrains them between 0 and 1. This enables a comprehensive evaluation of length, altitude, and horizontal turning angles as energy consumption costs.

(4) *Radar cost* $P_{i,R}$

The radars cost is calculated by the distance between control node of the UAV to the radar threat center and is described as follows:

$$P_{i,R} = \frac{1}{R}\cdot\sum_{r=1}^{R} LR_r \cdot \sum_{l=1}^{L}\left(\frac{1}{d_{i,l,r}^2}\right) \tag{5}$$

where R represents the number of radars, L represents the number of control nodes, LR_r represents the level of the radar r's threat, and $d_{i,l,r}$ represents the distance from control node l of the UAV i to the radar threat center r.

(5) *Anti-aircraft guns cost* $P_{i,G}$

The anti-aircraft guns cost is calculated by the distance between control node of the UAV to the anti-aircraft guns threat center and is described as follows:

$$P_{i,G} = \frac{1}{G}\cdot\sum_{g=1}^{G} LG_g \cdot \sum_{l=1}^{L}\left(\frac{1}{d_{i,l,g}^2}\right) \tag{6}$$

where G represents the number of anti-aircraft guns, LG_g represents the level of the anti-aircraft gun g's threat, and $d_{i,l,g}$ represents the distance from control node l of the UAV i to the anti-aircraft guns threat center g.

(6) *Missiles cost* $P_{i,M}$

The missiles cost is calculated by the distance between control node of the UAV to the missiles threat center and is described as follows:

$$P_{i,M} = \frac{1}{M}\cdot\sum_{m=1}^{M} LM_m \cdot \sum_{l=1}^{L}\left(\frac{1}{d_{i,l,m}^2}\right) \tag{7}$$

where M represents the number of missiles, LM_m represents the level of missile m's threat, and $d_{i,l,m}$ represents the distance from control node l of the UAV i to the missiles threat center m.

(7) *Geological hazard cost* $P_{i,H}$

The geological hazard cost is calculated as follows:

$$P_{i,H} = \frac{1}{H}\cdot\sum_{h=1}^{H} LH_h \cdot \sum_{l=1}^{L}\left(\frac{1}{d_{i,l,h}^2}\right) \tag{8}$$

where H represents the number of geological hazard threat areas, L represents the number of control nodes, LH_h represents the level of the geological hazard threat area h's threat, and $d_{i,l,h}$ represents the distance from control node l of the UAV i to the geological hazard threat area's center h.

3.3.2. Constraints for multiple UAVs flight

Multiple UAVs need to satisfy numerous constraints for safe flight, and collaborative path planning among multiple UAVs also needs to adhere to collision avoidance constraints. These constraint conditions are represented as follows:

$$\alpha_{i,\min} \le \alpha_{i,j} \le \alpha_{i,\max}, \forall i \in \{1,2,\ldots,I\} \tag{9}$$

$$(x_{i,j}, y_{i,j}, z_{i,j}) \le posbound_{i,\max} \tag{10}$$

$$(x_{i,j}, y_{i,j}, z_{i,j}) \ge posbound_{i,\min} \tag{11}$$

$$z_{i,j} > z', \text{ if } d_{i,j,b} \le d'_{\text{safe},b} \tag{12}$$

$$z_{i,l} > z_r, \text{ if } d_{i,l,r} \le d_r \tag{13}$$

$$z_{i,l} > z_g, \text{ if } d_{i,l,g} \le d_g \tag{14}$$

$$z_{i,l} > z_m, \text{ if } d_{i,l,m} \le d_m \tag{15}$$

$$z_{i,l} > z_h, \text{ if } d_{i,l,h} \le d_h \tag{16}$$

$$t_{kq} \ge t_{\text{safe}}, \text{ if } dis_{k,q} \le dis_{\text{safe}} \tag{17}$$

Among them, for UAV i, formula (9) represents the constraint on the horizontal turning angle when UAV i is flying, where $\alpha_{i,\min}$ and $\alpha_{i,\max}$ separately represent the minimum and maximum turning angles. Formula (10) is the maximum boundary constraint of the flight path, where $posbound_{i,\max} = (X_{i,\max}, Y_{i,\max}, Z_{i,\max})$ is the maximum values in three coordinates. Formula (11) stands for the minimum boundary constraint of the UAV's flight path, where $posbound_{i,\min} = (X_{i,\min}, Y_{i,\min}, Z_{i,\min})$ is the minimum values in three coordinates. Formula (12) presents the non-collision constraint between UAV i and obstacle, where z' is the height of the obstacle b. $d_{i,j,b}$ represents the horizontal projection of the distance between path node j of the UAV i and obstacle b. $d'_{\text{safe},b}$ represents the horizontal projection of the smallest safe distance between path node of the UAV and obstacle b. Formula (13) shows the threat constraint on the UAV's flight path from the radar center, where z_r represents the longitudinal range of radar threats, $d_{i,l,r}$ represents the horizontal projection of the distance between control node l of the UAV i and the radar center r, d_r represents the horizontal projection of the safe distance between control node l of the UAV's path and the radar center r. Formula (14) illustrates the threat constraint on the UAV's flight path from the anti-aircraft gun center, where z_g represents the longitudinal range of anti-aircraft gun threats, $d_{i,l,g}$ represents the horizontal projection of the distance between control node l of the UAV i and the anti-aircraft gun center g, d_g represents the horizontal projection of the safe distance between control node l of the UAV's path and the anti-aircraft gun center g. Formula (15) represents the threat constraint on the UAV's flight path from the missile center, where z_m represents the longitudinal range of missile threats, $d_{i,l,m}$ represents the distance between control node l of the UAV i and the missile center, d_m represents the horizontal projection of the safe distance between control node l of the UAV's path and the missile center m. Formula (16) shows the threat constraint on the UAV's flight path from the geological hazard center, where z_h represents the horizontal projection of the longitudinal range of geological hazard threats, $d_{i,l,h}$ represents the distance between control node l of the UAV i and the geological hazard center, d_h represents the horizontal projection of the safe distance between control node l of the UAV's path and the geological hazard center h. Formula (17) portraits the collaborative constraint between two UAVs, where t_{kq} represents the time difference between UAV k and q reaching any two waypoints, t_{safe} represents the safe time, dis_{kq} represents the distance between the two UAVs, and dis_{safe} represents the safe distance between the two UAVs.

4. The proposed TSKAC-NSGA-II algorithm

To optimize complex CMOPs, it is essential to balance the convergence and diversity of the solutions. Prioritizing convergence may result in accurate solutions but lead to overlapping points or similar solutions on the Pareto front. Emphasizing diversity alone could scatter solutions on the Pareto front but get some solutions with lower accuracy. To deal with this problem, the two-stage algorithm framework, where the first stage aims to enhance the convergence of the main population while the second stage focuses on the diversity of dominant particles in the dominant subpopulation, attracts our attention.

Based on the two-stage framework, the TSKAC-NSGA-II algorithm is proposed, where the division of stages is designed in formula (18).

$$Stage = \begin{cases} 1, & \text{if } iteration \leq \dfrac{MaxIt^2}{Z} \cdot \kappa, \\ 2, & \text{if } iteration < \dfrac{MaxIt^2}{Z} \cdot \kappa. \end{cases} \tag{18}$$

with

$$MaxIt_1 + MaxIt_2 = MaxIt$$

where κ is the division factor for different stages. $iteration$ is the current iteration number. Z is the number of chromosomes. $MaxIt_1$ is the maximum number of iterations for stage 1 and $MaxIt_2$ is the maximum number of iterations for stage 2. The framework of the designed algorithm is presented in Fig. 1.

4.1. The first stage

The first stage involves the coevolution of two populations: the main population and the assistant population. The main population optimizes the original problem, while the assistant population focuses solely on an assistant problem related to Obj_1, with no threat areas or obstacles in environment. Below provides the model.

$$\text{Assistant Problem} \begin{cases} \min & Obj_{i,1} \\ s.t. & (9), (10), (11), (17) \end{cases} \tag{19}$$

Due to the small feasible space of the original problem, the main population may get trapped in local regions, whereas the assistant problem has a larger feasible space with faster convergence speed. By sharing offspring information between these two populations, the assistant population can accelerate the convergence of the main population and help the main population escape from local optima.

Due to the complexity of the considered problems, searching in different directions and ensuring solution diversity are absolutely crucial. In this study, the population in the first stage and the dominant subpopulation in the second stage both employ NSGA-II algorithm. After obtaining a new parent population in each iteration, the individuals are sorted in ascending order based on the value of Obj_1, and the top 20% of the individuals adopt random exploration to update their positions, which aims to preserve the better parent individuals and allows those far from the actual UAV flight requirements to explore randomly in space, increasing the likelihood of discovering better solutions. The evolution expression is shown as follows:

$$X_l = X_l + q \cdot (posbound_{max} - posbound_{min}) \tag{20}$$

where q is an uniformly distributed value between 0 and 1. X_l is the position coordinates of the control nodes. $posbound_{max}$ and $posbound_{min}$ are the upper and lower bounds of the position coordinates of the control nodes of the UAVs, respectively.

4.2. The second stage

4.2.1. Double population coevolution

In the second stage, coevolution between two populations continues. The main population obtained from the first stage is divided into the dominant subpopulation and the inferior subpopulation based on the performance regarding Obj_1. Over iterations, the dominant subpopulation adopts the position update method of the GA algorithm, while the inferior subpopulation, within the QL framework, selects the best position update action from three different action modes based on multi-mode strategy. The number of the particles dynamically increases in the former while decreases in the latter. This stage primarily optimizes solutions in the inferior subpopulation and is treated as a whole entity. The most suitable population update mode is chosen from multiple behavioral modes based on the designed Q-learning (QL). After each iteration, the subpopulations are reorganized until the termination conditions are met.

4.2.2. Introduction about Q-learning

RL is an important field in machine learning. It allows an agent to make appropriate decisions by continuously exploring and learning from interacting with a given environment. QL is one of the RL methods, which includes five components: agent, action, environment, reward, and state. During the interaction with the environment, the agent continuously gains experience by trying out different actions and utilizes the obtained rewards and penalties to compute and update Q value according to the Q function, and then adaptively selects the optimal actions. The evolution of the Q function is as follows:

$$Q(s,a) = (1-\rho) \cdot Q(s,a) + \rho \cdot (r + \phi \cdot \max Q(s', a')) \tag{21}$$

where $Q(s,a)$ represents the Q value in state s after taking action a. ρ is the learning rate, which is used to control the impact of the new data on $Q(s,a)$, and is adaptively reduced to effectively learn the search space. r is the immediate reward obtained after an action, ϕ is the discount factor, s' is the next state and a' is the best action to take in s'. In below multi-mode strategy design, both Q-table and reward r are calculated following Ref. [13], while both state s, action a, and the indicators are strategically proposed.

4.2.3. Multi-mode strategy structure

The structure of the multi-mode strategy for inferior subpopulation based on QL algorithm is shown in Fig. 2. In multi-mode strategy, the inferior subpopulation is treated as an agent, and its state s, defined based on Obj_1, represents one of the situations. The entire search space is considered as an interactive environment, and the position update mode, which is taken by this subpopulation and is transitioning from the current state to another, is referred to as an action a. In this study, the agent selects the best one among three position update actions at state s through interaction with the search space, and update the state to s'. In multi-mode strategy, there exist four states and three actions. Moreover, four states are separated based on two indicators. Below will introduce them in detail.

4.2.4. Two indicators in multi-mode strategy

To better understand the separation of four states shown in Fig. 3, here, we first introduce relative distance and set optimality as two indicators.

Relative distance is defined by calculating $\frac{c-a}{d-b}$ and two scenarios can be separated based on $relativedistance > 1$ or $relativedistance \leq 1$. Here, we refer to particles at the far left and far right of the dominant subpopulation as $domi_{best}$ and $domi_{bad}$ with values a and b on Obj_1, respectively. Similarly, for the inferior subpopulation, particles at the far left and far right are denoted as $inferi_{best}$ and $inferi_{bad}$, with values c and d on Obj_1, respectively.

Set optimality is defined by calculating $\frac{num_{fo}}{num_{la}}$ and two scenarios can be separated based on $setoptimality > 1$ or $setoptimality \leq 1$. As shown in Fig. 3, the inferior subpopulation is divided into yellow and purple portions by $\frac{c+d}{2}$. The number of particles in the yellow and purple portions are denoted as num_{fo} and num_{la}, respectively.

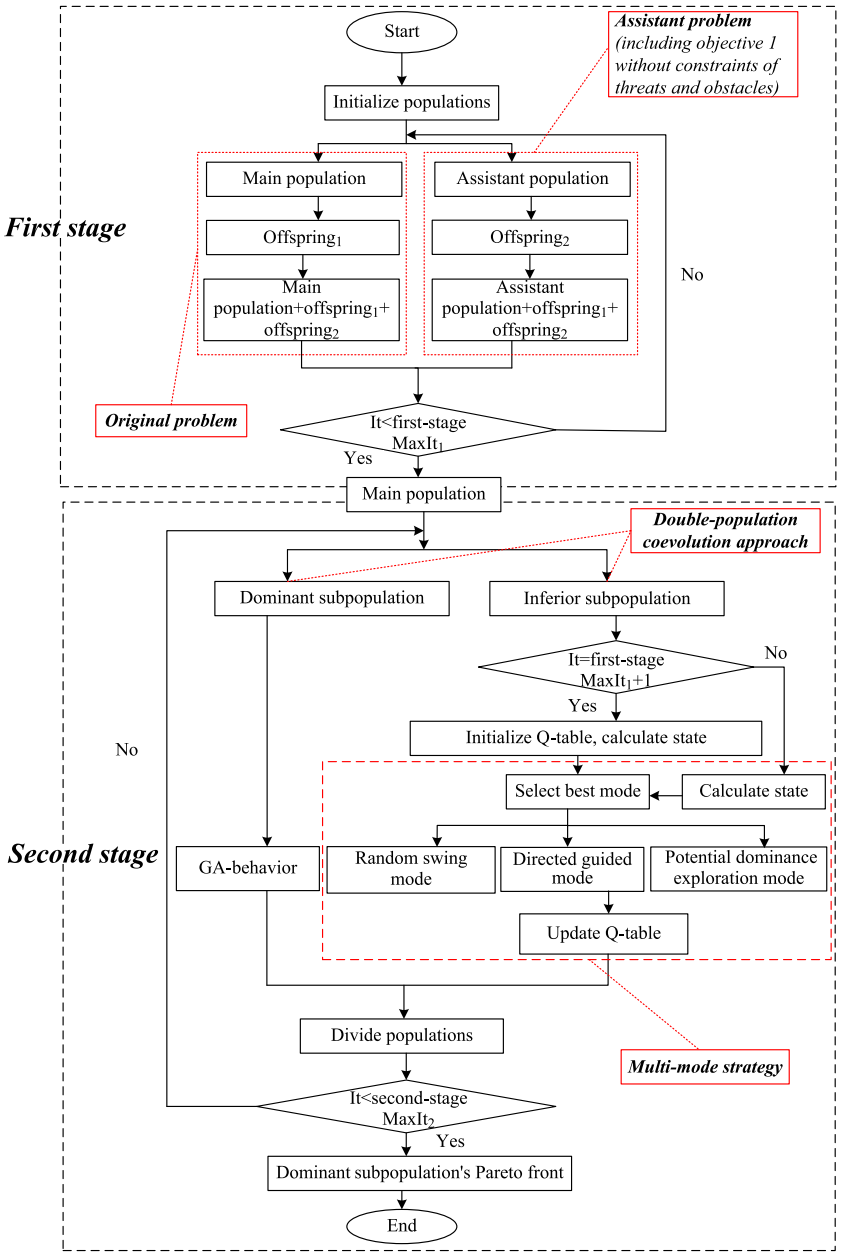

Fig. 1. The framework of TSKAC-NSGA-II algorithm.

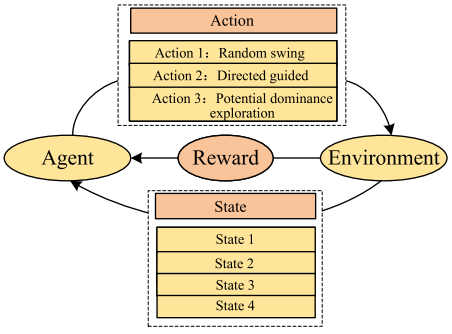

Fig. 2. Multi-mode strategy for inferior subpopulation based on Q-learning algorithm.

4.2.5. Four states in multi-mode strategy

In multi-mode strategy described in Fig. 2, four states are separated based on relative distance and set optimality.

States 1 is depicted in Fig. 3(a), where $relativedistance > 1$ and $setoptimality > 1$. The inferior subpopulation is relatively narrow compared to the dominant subpopulation, and particles are concentrated in the front half, with more good particles than bad particles.

State 2 is depicted in Fig. 3(b), where $relativedistance \leq 1$ while $setoptimality > 1$. The inferior subpopulation is relatively wide compared to the dominant subpopulation, and particles are concentrated in the front half, with more good particles than bad particles.

State 3 is depicted in Fig. 3(c), where $relativedistance > 1$ while $setoptimality \leq 1$. The inferior subpopulation is relatively narrow compared to the dominant subpopulation, and particles are concentrated in the rear half, with more bad particles than good particles.

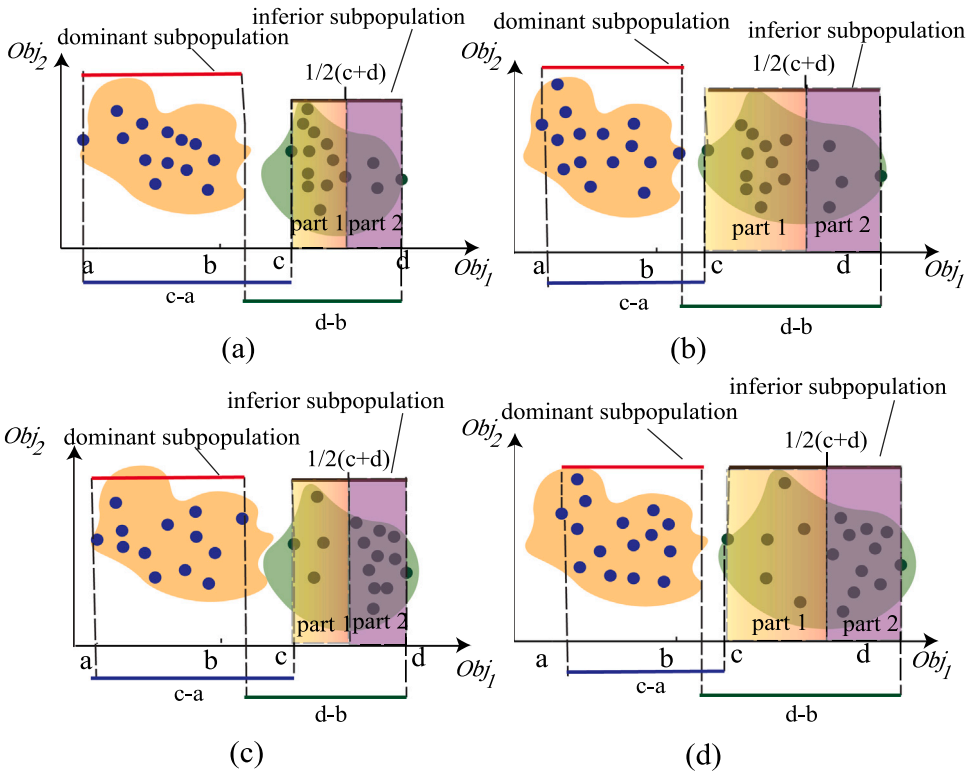

Fig. 3. Four states of the inferior subpopulation in multi-mode strategy.

State 4 is depicted in Fig. 3(d), where $relativedistance \leq 1$ and $setoptimality \leq 1$. The inferior subpopulation is relatively wide compared to the dominant subpopulation, and particles are concentrated in the rear half, with more bad particles than good particles.

4.2.6. Three actions in multi-mode strategy

In multi-mode strategy described in Fig. 2, three actions are designed.

Action 1 is the random swing mode. This mode is used to expand the coverage of the search space and help individuals escape from local optima. Within the feasible domain, individuals undergo random oscillations in order to increase the likelihood of finding better solutions. As the number of iterations increases, the magnitude and range of this search will decrease. The main idea of this method is to introduce a certain level of randomness to assist individuals in the inferior subpopulation in exploring a wider feasible domain, thereby increasing the likelihood of discovering superior solutions. Specifically, a random term within a specific range is added to each dimension of the original solution. The expression for this method is described as follows:

$$X_l = X_l + (-1)^r \cdot \gamma \cdot (1 - \frac{iteration}{MaxIt}) \cdot (posbound_{max} - X_l) \tag{22}$$

where X_l is the individual l's original location, r is a random number following a binomial distribution with parameters 0 and 1. γ is a random number following a uniform distribution from 0 to 1, and $posbound_{max}$ is the upper bound of the solution.

Action 2 is the directed guided mode. This mode updates the positions of individuals in the inferior subpopulation by utilizing the mean of the dominant subpopulation as a reference point to guide the former towards more promising directions. This approach aims to leverage the information from the dominant subpopulation to steer the entire population towards better exploration, thereby improving search efficiency. The expression is described as follows:

$$X_l = f \cdot X_l + h \cdot dom_{mean} \tag{23}$$

with

$$f + h = 1$$

where f is a random number following a uniform distribution from 0 to 1, dom_{mean} is the mean value of the dominant subpopulation.

Action 3 is the potential dominance exploration mode. quantum-behaved particle swarm optimization (QPSO) algorithm [48] eliminates the directional properties of the particle movement, allowing particles globally influenced by all particles, including lagged ones. This strategy can help particles explore potential optimal solutions in a broader region, so that the population can escape from local optimal solutions. Inspired by this idea, the individuals in the inferior subpopulation are allowed to be influenced by all individuals in the superior subpopulation, enabling exploration around the superior subpopulation. This is defined as potential dominance exploration, and the expression is listed as follows:

$$X_l = P_l + \epsilon \cdot |MV - X_l| \cdot \ln \frac{1}{\mu} \tag{24}$$

with

$$MV = \frac{1}{L'} \sum_{l=1}^{L'} X_l$$

$$P_l = \xi \cdot X_l + (1 - \xi) \cdot Rbest$$

where P_l is local attractor. ϵ is the coefficient creativity, and μ and ξ are uniformly distributed values between 0 and 1. MV is the mean value of individuals in inferior subpopulation, $Rbest$ is a random individual from dominant subpopulation, and L' is the pop size of inferior subpopulation.

4.3. Path planning for multiple UAVs based on TSKAC-NSGA-II algorithm

For the multiple UAVs path planning problem, we first consider the path planning for a single UAV. Then, we perform collision detection to achieve coordination among multiple UAVs. The pseudo code of path planning for multiple UAVs is shown in Algorithm 1.

4.4. Computing complexity

num_UAV represents the number of UAVs. The number of populations in the first stage and the number of subpopulations in the second stage are both 2. Z denotes the number of individuals (chromosomes), and Ob is the number of objectives. Considering the multiple UAVs path planning model for the algorithm [13], the proposed TSKAC-NSGA-II algorithm has the following time complexities for its two stages:

4.4.1. The first stage

In the first stage, the path optimization model for each UAV is independently solved by 2 populations, with each population containing Z individuals, all of which evolve according to the NSGA-II methodology. Consequently, the time complexity of the first stage is $O\left(MaxIt_1 \cdot num\_UAV \cdot 2 \cdot Ob \cdot Z^2\right) = O\left(MaxIt_1 \cdot num\_UAV \cdot Ob \cdot Z^2\right)$

4.4.2. The second stage

The second stage is further divided into the dominant subpopulation and the inferior subpopulation. The dominant subpopulation updates in a singular manner, while the inferior subpopulation selects one of three different updating strategies in each generation, both employing the same non-dominated sorting approach. Therefore, considering both subpopulations uniformly, the time complexity of the second stage is $O\left(MaxIt_2 \cdot num\_UAV \cdot Ob \cdot Z^2\right)$.

5. Experimental studies

To evaluate adaptability and performance of our proposed TSKAC-NSGA-II algorithm to different environments and various geographical parameters, we conducted a series of simulation experiments in two application scenarios. The experimental platform was a personal computer with the Windows 11 operating system, powered by an AMD Ryzen R5-6600H processor and equipped with 16 GB of RAM.

We compared the proposed TSKAC-NSGA-II algorithm with five other algorithms in two experimental environments, including MOPSO [49], NSGA-II [50], MCMOPSO-RL [13], TS-NSGA-II [51] and CCMO [44]. Each algorithm independently conducted 10 multiple UAVs path planning experiments in each environment. Through a large number of preliminary experiments, the main parameters for the TSKAC-NSGA-II algorithm are set as follows: maximum number of iterations $t_{max} = 800$, population size $npop = 80$, crossover probability $crp = 0.8$, mutation probability $mutp = 1/12$, division factor for stages $\kappa = 0.06$, discount factor $\phi = 0.5$ and energy consumption factors $\beta_1 = 100$, $\beta_2 = 50$, $\beta_3 = 20$.

5.1. Parameter analysis experiment

To assess the impact of parameters on algorithm performance and determine optimal combinations, this section performs variance analysis on three critical parameters: division factor κ, initial learning rate ρ_1, and final learning rate ρ_2. κ takes values of 0.03 (short stage one), 0.06 (medium stage one), and 0.09 (long stage one). Given the constraint that learning rates range between 0 and 1, and considering their effect across iterations, the analysis sets $\rho_1 + \rho_2 = 1$. Three configurations are examined: a constant and balanced rate ($\rho_1 = \rho_2 = 0.5$), an increasing rate ($\rho_1 = 0.9, \rho_2 = 0.1$), and a decreasing rate ($\rho_1 = 0.1, \rho_2 = 0.9$). In practice, the analysis focuses solely on the influence of κ and ρ_1 values on the algorithm. Fig. 4 illustrates the TSKAC-NSGA-II algorithm's performance on total Obj_1 across 9 parameter combinations in environment 1. Table 4 presents the outcomes of 10 runs of the TSKAC-NSGA-II algorithm for total Obj_1 under these parameter settings. Table 5 summarizes the algorithm's best, median, mean, worst values, and standard deviation across different parameter combinations in environment 1.

Algorithm 1 Pseudocode of path planning for multiple UAVs based on TSKAC-NSGA-II algorithm

Initialize: establish a simulated map; num_UAV: number of UAVs; $\tilde{L}$: number of control points; L: number of control nodes; Z: number of chromosomes for each population; X_z: individual's position; $MaxIt$: max iterations; crp: crossover probability; $mutp$: mutation probability; ϕ: discount factor; main population and assistant population; τ: stage control factor; κ: division factor for stages; λ_1, λ_2, λ_3, λ_4: threat area factors; β_1, β_2, β_3: energy consumption factors.

for $iteration = 1{:}MaxIt_1$ **do**
  for $nuav = 1{:}num\_UAV$ **do**
    $Offspring_1$: Generate offspring from main population;
    $Offspring_2$: Generate offspring from assistant population;
    Main population = main population$\cup Offspring_1 \cup Offspring_2$;
    Assistant population = assistant population$\cup Offspring_1 \cup Offspring_2$;
    Calculate fitness value of both populations;
    Sort the population in descending order according to Obj_1;
    if X_z is in the top 20% of each population **then**
      Update X_z by Equ. (20);
    end if
  end for
end for
Output:
main population of each UAV
divide main population into dominant and inferior subpopulations;
for $iteration = 1{:}MaxIt_2$ **do**
  for $nuav = 1{:}num\_UAV$ **do**
    for $o = 1{:}2$ **do**
      if $o = 1$ **then**
        Update X_z for dominant subpopulation based on the approach of evolutionary method of the first stage;
      else
        Calculate state of inferior subpopulation;
        Switch position update action
          Case 1: directed guided mode by Equ. (23);
          Case 2: random swing mode by Equ. (22);
          Case 3: potential dominance exploration mode by Equ. (24);
        Calculate reward r;
        Update the Q-table;
      end if
      Calculate fitness value;
    end for
    Re-merge the two subpopulations into a population and divide population into dominant and inferior subpopulations;
  end for
end for
Output:
dominant subpopulations of each UAV.

Table 1
Start and destination positions in environment 1.

UAV	Start position			Destination position		
	X	Y	Z	X	Y	Z
1	5	4	30	90	95	60
2	47	1	35	6	92	70
3	92	5	25	43	94	50

Table 2
Locations of the buildings in environment 1.

Building	X range	Y range	Height (m)
1	10–25	10–25	60
2	60–80	60–80	30
3	20–30	80–90	100
4	7–22	40–60	80
5	28–52	10–30	90

Table 3
Threat areas in environment 1.

Threat	Center	Radius	Height (m)
Radar	(35,60)	12	30
Anti-aircraft gun	(70,20)	12	20
Missile	(60,45)	10	35

Table 4
10 runs of TSKAC-NSGA-II for total Obj_1 under 9 parameter combinations in environment 1.

(κ, ρ_1, ρ_2)	Results				
(0.03, 0.5, 0.5)	38.38	38.10	38.03	37.99	37.96
	37.71	37.69	37.64	37.61	37.60
(0.03, 0.9, 0.1)	45.34	44.88	38.57	38.08	37.72
	37.64	37.55	37.45	37.44	37.45
(0.03, 0.1, 0.9)	48.32	43.05	41.18	40.75	40.37
	37.78	37.59	37.54	37.53	37.52
(0.06, 0.5, 0.5)	40.57	38.12	37.19	37.13	37.12
	37.11	37.11	37.01	37.09	37.08
(0.06, 0.9, 0.1)	45.21	44.85	38.84	38.28	37.87
	37.14	**36.93**	**36.43**	**36.44**	**36.50**
(0.06, 0.1, 0.9)	39.90	38.99	37.79	37.69	37.54
	37.48	37.46	37.46	37.46	37.46
(0.09, 0.5, 0.5)	41.97	41.68	39.99	39.72	39.59
	38.83	38.71	39.01	39.00	38.99
(0.09, 0.9, 0.1)	44.48	40.82	41.42	41.19	41.04
	40.61	40.21	39.94	40.27	39.59
(0.09, 0.1, 0.9)	42.34	38.92	39.37	39.57	39.24
	39.11	39.29	39.32	39.44	39.24

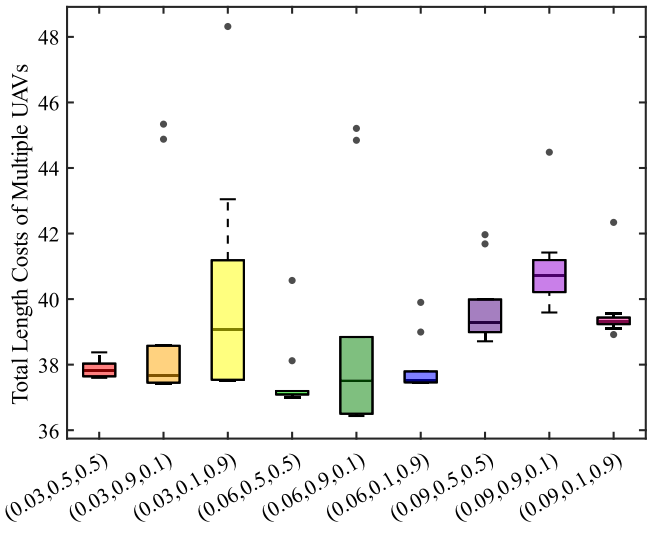

Fig. 4. Boxplot of 10 trials for 9 parameter combinations in environment 1.

Table 5
Performance comparisons of 9 parameter combinations'TSKAC-NSGA-II algorithm in environment 1.

	(ρ_1, ρ_2)	Best	Median	Mean	Worst	Std
$\kappa = 0.03$	$(\rho_1 = 0.5, \rho_2 = 0.5)$	37.60	37.84	37.87	**38.38**	**0.26**
	$(\rho_1 = 0.9, \rho_2 = 0.1)$	37.44	37.68	39.21	45.34	3.13
	$(\rho_1 = 0.1, \rho_2 = 0.9)$	37.52	39.07	40.16	48.32	3.49
$\kappa = 0.06$	$(\rho_1 = 0.5, \rho_2 = 0.5)$	37.01	**37.12**	**37.55**	40.57	1.11
	$(\rho_1 = 0.9, \rho_2 = 0.1)$	**36.43**	37.51	38.85	45.21	3.36
	$(\rho_1 = 0.1, \rho_2 = 0.9)$	37.46	37.51	37.92	38.90	0.84
$\kappa = 0.09$	$(\rho_1 = 0.5, \rho_2 = 0.5)$	38.71	39.30	39.75	41.97	1.17
	$(\rho_1 = 0.9, \rho_2 = 0.1)$	39.59	40.71	40.96	44.48	1.36
	$(\rho_1 = 0.1, \rho_2 = 0.9)$	38.92	39.31	39.58	42.34	0.98

Table 1 displays the start position and destination positions of UAVs in environment 1, Table 2 illustrates the location information of buildings and Table 3 shows the threat areas.

Table 6
Performance comparisons of different algorithms for multiple UAVs in environment 1.

Algorithm	Best	Median	Mean	Worst	Std
MOPSO [49]	58.53	92.95	92.48	133.76	19.83
NSGA-II [50]	60.48	71.84	72.76	87.53	9.30
MCMOPSO-RL [13]	118.06	131.75	134.40	169.71	15.52
TS-NSGA-II [51]	45.17	57.63	64.71	99.99	17.35
CCMO [44]	44.71	49.76	51.24	59.55	4.19
TSKAC-NSGA-II	**36.43**	**37.87**	**38.85**	**45.21**	**3.36**

Table 7
Performance comparisons of different algorithms for each UAV in environment 1.

	Algorithm	Best	Median	Mean	Worst	Std
UAV1	MOPSO [49]	12.78	23.95	27.23	48.94	10.05
	NSGA-II [50]	15.46	20.68	20.19	24.58	2.80
	MCMOPSO-RL [13]	27.55	40.29	38.18	50.12	6.56
	TS-NSGA-II [51]	12.85	19.80	18.04	22.56	3.96
	CCMO [44]	15.05	25.05	18.08	22.47	2.58
	TSKAC-NSGA-II	**11.48**	**12.01**	**12.01**	**12.59**	**0.41**
UAV2	MOPSO [49]	25.16	38.02	44.07	61.42	11.56
	NSGA-II [50]	27.34	32.04	33.05	39.87	4.42
	MCMOPSO-RL [13]	34.35	48.48	49.60	65.08	11.18
	TS-NSGA-II [51]	20.77	28.58	29.34	41.45	6.53
	CCMO [44]	20.94	25.05	25.54	31.83	3.22
	TSKAC-NSGA-II	**17.86**	**18.42**	**19.72**	**25.17**	**2.92**
UAV3	MOPSO [49]	9.24	13.91	21.18	57.15	16.91
	NSGA-II [50]	14.64	16.82	19.52	27.09	5.14
	MCMOPSO-RL [13]	32.48	47.32	46.62	54.51	6.23
	TS-NSGA-II [51]	7.94	9.11	17.32	56.60	17.76
	CCMO [44]	7.27	7.70	7.61	8.16	0.28
	TSKAC-NSGA-II	**7.05**	**7.08**	**7.13**	**7.49**	**0.13**

From Table 4 and Fig. 4, it is evident that the TSKAC-NSGA2 algorithm with the fifth parameter set performs better in finding solutions with smaller values for the objective function 1, outperforming other parameter combinations overall. The data in Table 5 clearly shows that although the algorithm with the first parameter set exhibits greater stability. However, its overall performance is poor. The algorithm with the fourth parameter set is generally better, but it is unable to find solutions with smaller values for Obj_1. The algorithm with the fifth parameter set has a distinct advantage in locating solutions with even smaller values for Obj_1, indicating its higher potential to discover optimal UAVs' flight paths. Consequently, this study selects the fifth parameter set as the recommended configuration for the TSKAC-NSGA-II algorithm and will proceed with it as the basis for subsequent simulation experiments.

5.2. Simulation experiments for multiple UAVs in environment 1

Environment 1 consists of five buildings, one radar threat, one anti-aircraft gun threat, and one missile threat. For each UAV in environment 1, the maximum horizontal turning angle limit $\alpha_{\max} = 55°$, the threat levels for radar, gun, and missile threats $LR_r = LG_g = LM_m = 10$, the geological hazard threat $LH_h = 0$. The flight speed $v = 4$ m/s, threat area factors $\lambda_1 = \lambda_2 = \lambda_3 = 1$, $\lambda_4 = 0$ and the ranges for flight environments are $100\,\text{m} \times 100\,\text{m} \times 100\,\text{m}$. Fig. 5 shows the Pareto fronts obtained by running six algorithms ten times to find the optimal flight path in environment 1. Figs. 6–8 show the side, top and front views of flight paths, respectively.

Table 6 lists the best, median, mean, worst values, and standard deviation of Obj_1 for the six algorithms in environment 1. Table 7 lists the best, median, mean, worst values, and standard deviation of Obj_1 for each UAV in environment 1, associated with the six algorithms.

It can be seen from Fig. 7 that the TSKAC-NSGA-II algorithm plans the shortest path for three UAVs. From Fig. 6, it plans smoother paths, minimizing times and turning magnitude as well as vertical ascents or descents. Fig. 8 shows that our proposed algorithm smoothly navigates buildings, avoids threat areas, and ensures that the paths are closest to a straight line, connecting the starting and destination positions. Fig. 5

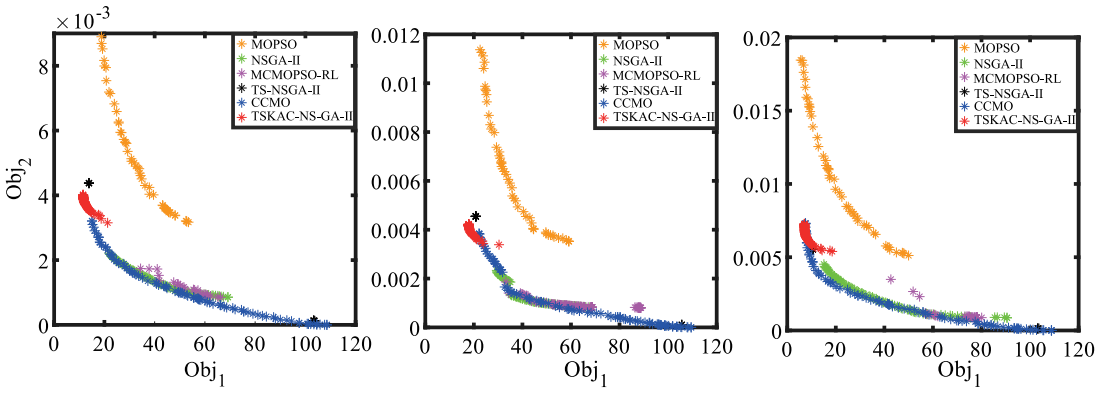

Fig. 5. Pareto fronts of various algorithms in environment 1.

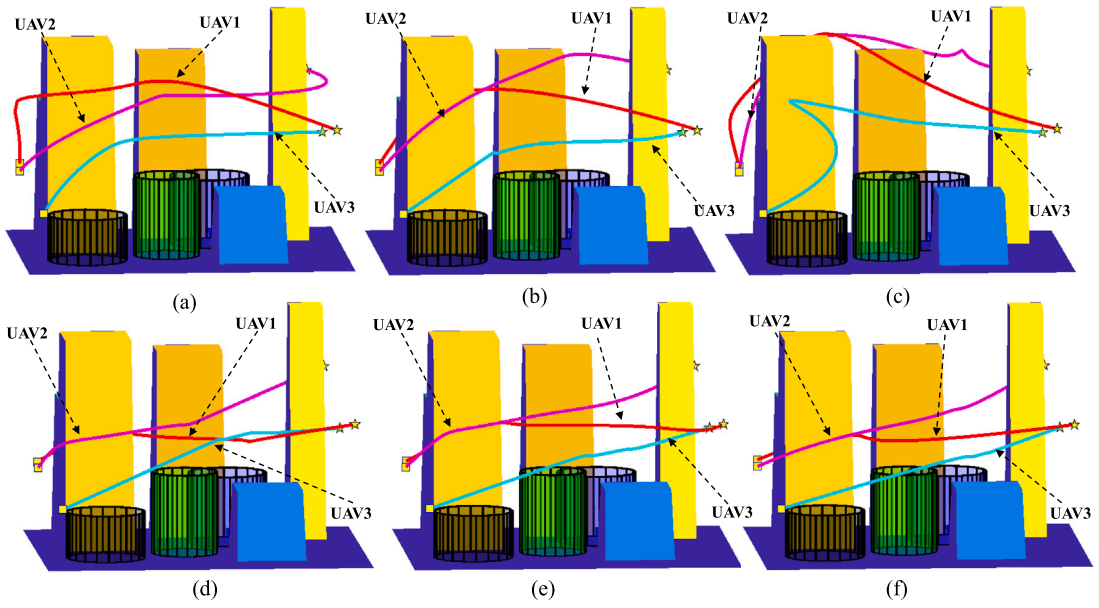

Fig. 6. Paths of various algorithms in environment 1 in a side view. (a) MOPSO. (b) NSGA-II. (c) MCMOPSO-RL. (d) TS-NSGA-II. (e) CCMO. (f) TSKAC-NSGA-II.

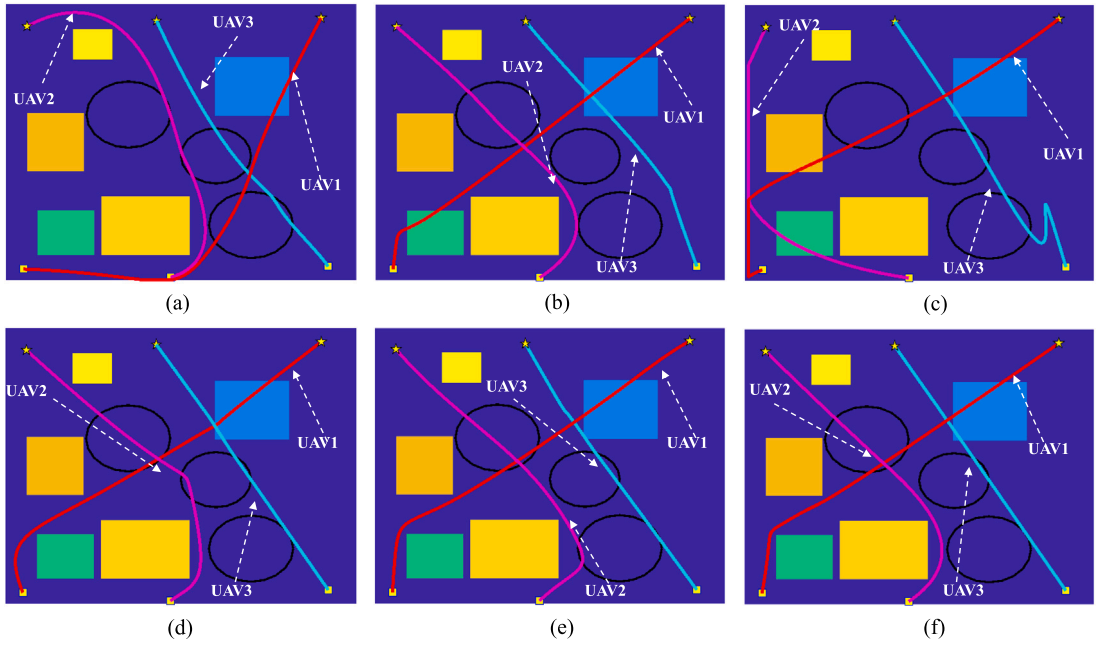

Fig. 7. Paths of various algorithms in environment 1 in a top view. (a) MOPSO. (b) NSGA-II. (c) MCMOPSO-RL. (d) TS-NSGA-II. (e) CCMO. (f) TSKAC-NSGA-II.

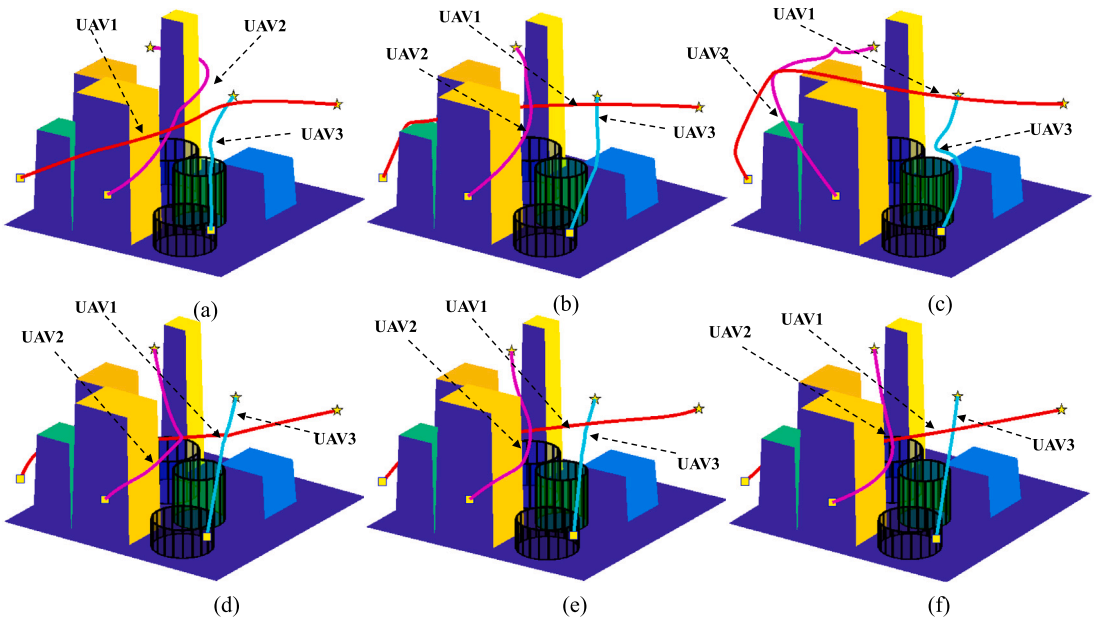

Fig. 8. Paths of various algorithms in environment 1 in a front view. (a) MOPSO. (b) NSGA-II. (c) MCMOPSO-RL. (d) TS-NSGA-II. (e) CCMO. (f) TSKAC-NSGA-II.

shows that the Pareto fronts of our algorithm is closer to the left and shorter, which means that Obj_1 of our algorithm is smaller, resulting in lower energy consumption for the alternative paths of the UAVs compared with other algorithms. This indicates that our algorithm exhibits better convergence in this environment and can escape from local optima.

As shown in Table 6, the TSKAC-NSGA-II algorithm outperforms other algorithms (almost twice better compared with the worst comparison algorithm) by a significant margin in terms of the minimum value, median, mean, maximum value, and standard deviation of the total sum of Obj_1 for the planned paths. Moreover, it can be seen from Table 7 that the Obj_1 observed by TSKAC-NSGA-II algorithm for each UAV also ranks the first (almost twice better compared with the worst comparison algorithm) compared with other algorithms in terms of the minimum, median, mean, and worst values. Furthermore, the standard deviation of the planned path for each UAV is also the lowest among various compared algorithms. This suggests that in the given environment 1, no matter for each UAV or UAVs, our algorithm not only exhibits good convergence for the planned paths but also demonstrates good stability.

Remark 3. The purpose of this study is to find flight paths with the minimum energy consumption outside the threat area. Therefore, we hope that the final alternative paths all have relatively low energy consumption. This means that, given Obj_2 is acceptable, smaller values of Obj_1 are preferred, because paths with larger Obj_1 do not have practical significance even if they satisfy the condition of avoiding collisions with other UAVs. This implies that the Pareto front should be closer to the left and as short as possible.

5.3. Simulation experiments for multiple UAVs in environment 2

Environment 2 is designed to simulate mountainous environment terrain, including several mountains and three threat areas. For each UAV in environment 2, the maximum horizontal turning angle limit $\alpha_{\max} = 55°$, the threat levels for threats $LR_r = LG_g = LM_m = 0$, $LH_1 = LH_2 = LH_3 = 10$. The flight speed $v = 6$ m/s, threat area factors $\lambda_1 = \lambda_2 = \lambda_3 = 0$, $\lambda_4 = 1$ and the ranges for flight environments are 200 m × 200 m × 200 m. The Pareto fronts for each algorithm are shown in Fig. 9. Fig. 10 illustrates the side view of the optimal flight paths obtained from six algorithms that run ten times respectively in environment 2, while the corresponding top and front views are depicted in Figs. 11 and 12, respectively.

Table 8
Start and destination positions in environment 2.

UAV	Start position			Destination position		
	X	Y	Z	X	Y	Z
1	12	10	60	180	190	120
2	94	2	70	12	184	140
3	184	10	54	86	188	110

Table 9
Threat areas in environment 2.

Threat	Center	Radius	Height (m)
Geological hazard area 1	(140,40)	30	40
Geological hazard area 2	(70,120)	30	60
Geological hazard area 3	(120,90)	20	80

Table 10
Performance comparisons of different algorithms for multiple UAVs in environment 2.

Algorithm	Best	Median	Mean	Worst	Std
MOPSO [49]	50.82	68.39	78.42	109.81	19.40
NSGA-II [50]	56.69	64.06	66.10	80.02	7.71
MCMOPSO-RL [13]	117.73	131.90	132.54	143.66	7.99
TS-NSGA-II [51]	89.05	108.79	125.75	178.32	32.58
CCMO [44]	33.21	35.95	35.95	47.49	4.60
TSKAC-NSGA-II	**26.16**	**26.34**	**26.48**	**27.33**	**0.36**

Table 8 displays the start position and destination positions of UAVs in environment 2, Table 9 shows the threat areas, Table 10 presents the best value, median, mean, worst value, and standard deviation of the total Obj_1 for six algorithms. Additionally, and Table 11 lists the best value, median, mean, worst value, and standard deviation of Obj_1 about each UAV for the six algorithms.

It can be seen from three perspectives of the figures, the paths planned by the TSKAC-NSGA-II algorithm can perfectly avoid terrain and threat areas, and are the shortest and smoothest compared with other algorithms. From the Pareto fronts in Fig. 11, it can be seen that under the condition where Obj_2 is less than a certain threshold, the Obj_1 of the TSKAC-NSGA-II algorithm is the most advanced, and our algorithm performs the best in the expectation of short, smooth, and

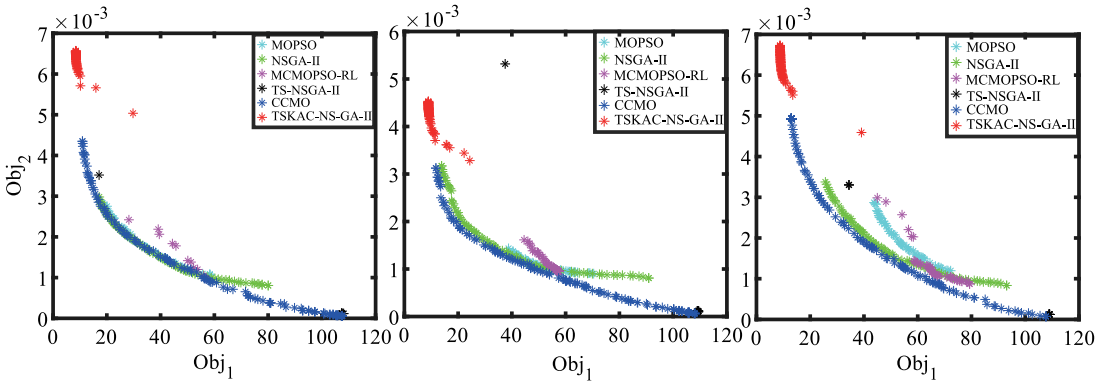

Fig. 9. Pareto fronts of various algorithms in environment 2.

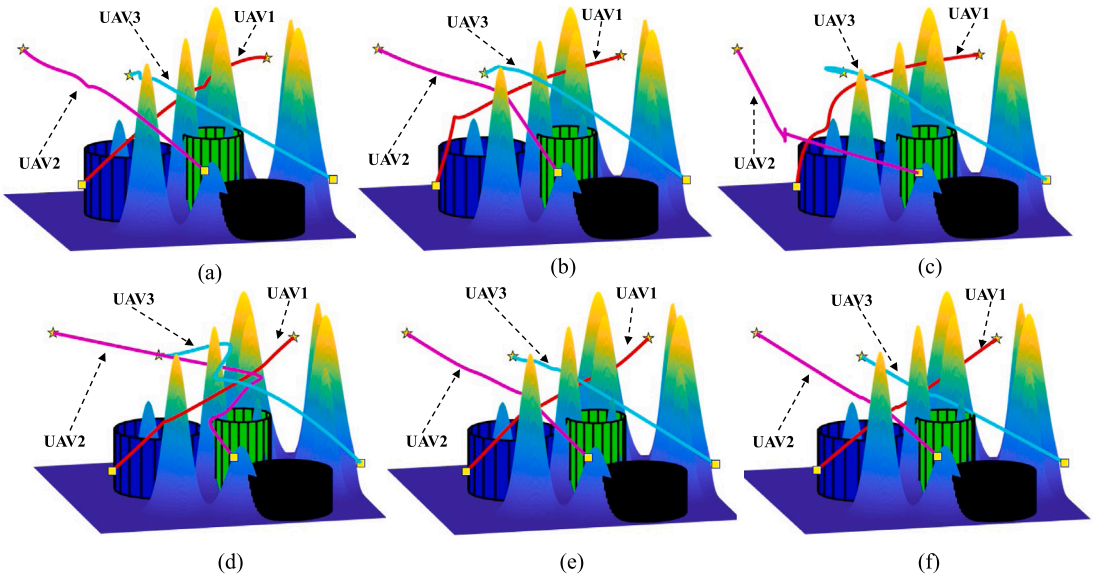

Fig. 10. Paths of various algorithms in environment 2 in a side view. (a) MOPSO. (b) NSGA-II. (c) MCMOPSO-RL. (d) TS-NSGA-II. (e) CCMO. (f) TSKAC-NSGA-II.

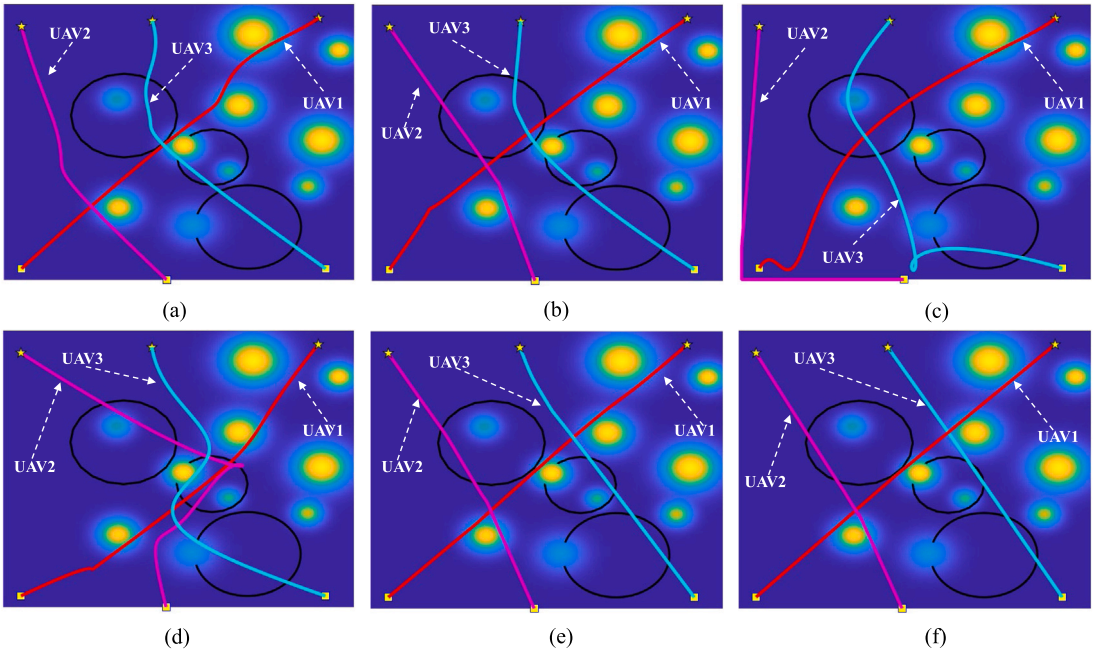

Fig. 11. Paths of various algorithms in environment 2 in a top view. (a) MOPSO. (b) NSGA-II. (c) MCMOPSO-RL. (d) TS-NSGA-II. (e) CCMO. (f) TSKAC-NSGA-II.

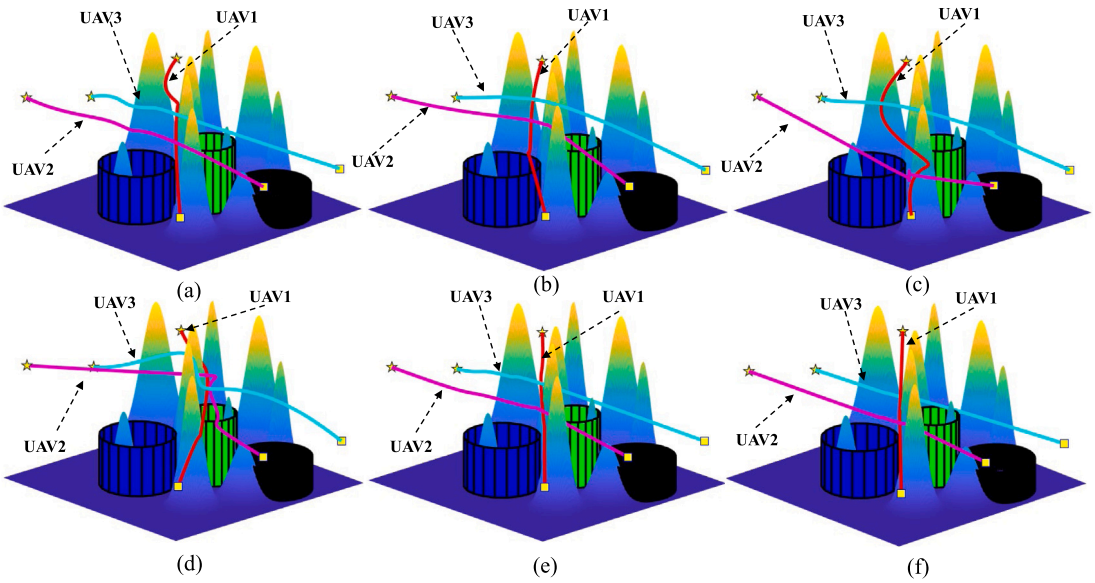

Fig. 12. Paths of various algorithms in environment 2 in a front view. (a) MOPSO. (b) NSGA-II. (c) MCMOPSO-RL. (d) TS-NSGA-II. (e) CCMO. (f) TSKAC-NSGA-II.

Table 11
Performance comparisons of different algorithms for each UAV in environment 2.

	Algorithm	Best	Median	Mean	Worst	Std
UAV1	MOPSO [49]	12.75	17.25	19.47	32.09	5.66
	NSGA-II [50]	14.38	16.75	16.31	18.52	1.62
	MCMOPSO-RL [13]	28.22	35.16	35.04	40.16	3.50
	TS-NSGA-II [51]	12.18	16.07	17.26	22.49	3.10
	CCMO [44]	10.01	10.77	10.81	11.98	0.55
	TSKAC-NSGA-II	**8.47**	**8.72**	**8.69**	**9.36**	**0.27**
UAV2	MOPSO [49]	15.56	24.15	30.14	47.51	11.57
	NSGA-II [50]	13.85	18.82	20.89	27.70	5.12
	MCMOPSO-RL [13]	33.92	38.79	39.80	44.55	3.49
	TS-NSGA-II [51]	18.76	37.38	49.27	110.90	33.03
	CCMO [44]	10.22	12.17	12.12	16.59	2.09
	TSKAC-NSGA-II	**8.92**	**8.95**	**8.95**	**8.98**	**0.02**
UAV3	MOPSO [49]	17.16	25.86	28.81	47.19	9.90
	NSGA-II [50]	21.12	29.15	28.90	37.29	4.49
	MCMOPSO-RL [13]	44.97	57.10	57.71	65.78	6.05
	TS-NSGA-II [51]	34.42	45.79	59.23	61.05	27.64
	CCMO [44]	12.29	13.66	15.11	20.88	3.28
	TSKAC-NSGA-II	**8.74**	**8.82**	**8.84**	**9.00**	**0.09**

minimal undulating paths. The paths planned by the TSKAC-NSGA-II algorithm for a single UAV have relatively small Obj_1, ensuring the efficiency of the cooperative path planning for multiple UAVs.

From the numerical results, Table 10 shows that the TSKAC-NSGA-II algorithm's minimum sum, median, mean, and maximum values of Obj_1 for the planned paths are far better (almost twice better compared with the worst comparison algorithm) than the other algorithms, and the standard deviation is also the smallest. Table 11 demonstrates that the TSKAC-NSGA-II algorithm's minimum, median, and mean value of Obj_1 for the path planned for each UAV is far superior (almost twice better compared with the worst comparison algorithm) to several other algorithms. This indicates that in environment 2, no matter for each UAV or UAVs, our algorithm not only exhibits good convergence but also demonstrates good stability.

The experimental results indicate that the TSKAC-NSGA-II algorithm performs better in terms of solution optimality compared to several other algorithms. In the simulated experiments in two different environments, our algorithm can obtain more desired UAV paths with shorter and better Obj_1 values. This demonstrates the algorithm has greater search performance and superior capability in solving multiple UAVs path planning problems compared to other algorithms.

6. Conclusion and future direction

This paper investigates the bi-objective path planning problem of multiple unmanned aerial vehicles. First, we model the single UAV path planning problem as a bi-objective model, where the first objective aims to obtain shorter, smoother, and more stable flight paths, while the second objective aims to minimize the threat for each path. Then, we propose the TSKAC-NSGA-II algorithm, where the first stage focuses on convergence, and the second stage emphasizes the diversity of solutions. Moreover, for the second stage, both double-population coevolution approach and the multi-mode strategy is introduced to choose the appropriate position update mode. Finally, two different environments are tested and five comparison algorithms are utilized to verify the superiority of our proposed algorithm.

Although the algorithm we proposed can generate a high-quality set of UAV paths, has demonstrated robustness in two distinct environments and is well-suited for the efficient operation of multiple UAVs, it still has certain limitations that require further in-depth study. In future research, we will consider rescue scenarios with more complex environments and different scales. Moreover, we will prioritize the development of dynamic obstacle avoidance mechanisms to ensure UAV flight safety. Additionally, we plan to design heuristic multi-objective optimization algorithms that are tailored to the specific challenges of our problem domain, aiming to enhance the algorithm's accuracy and efficiency.

CRediT authorship contribution statement

Tianwei Zhou: Writing – original draft, Supervision, Project administration, Methodology, Investigation. **Zhenghan Zhou:** Writing – original draft, Software, Methodology, Investigation, Conceptualization. **Haiyun Qiu:** Software, Investigation. **Ben Niu:** Writing – review & editing, Supervision, Project administration. **Gabriel Xiao-Guang Yue:** Supervision, Investigation. **Witold Pedrycz:** Writing – review & editing, Supervision.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No. 72334004, No. 62103286, No. 71971143, in part by Guangdong Provincial Philosophy and Social Sciences Planning Project under Grant No. GD22XGL22, No. GD22CGL35, in part by the Basic and Applied Basic Research Foundation of Guangdong Province under Grant No. 2024A1515030278, Grant No. 2024A1515011712, in part by Guangdong Province Innovation Team Intelligent Management and Interdisciplinary Innovation under Grant No. 2021WCXTD002, in part by Special Projects in Key Fields of Ordinary Colleges and Universities in Guangdong Province under Grant No. 2022ZDZX2054.

References

[1] R. Austin, Unmanned Aircraft Systems: UAVs Design, Development and Deployment, John Wiley & Sons, 2011.
[2] H. Nawaz, H.M. Ali, S. Massan, Applications of unmanned aerial vehicles: a review, Tecnol. Glosas Innov. Apl. Pyme. Spec. (2019) (2019) 85–105.
[3] B. Xu, K. Zhao, Q. Luo, G. Wu, W. Pedrycz, A GV-drone arc routing approach for urban traffic patrol by coordinating a ground vehicle and multiple drones, Swarm Evol. Comput. 77 (2023) 101246.
[4] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I.L. Grixa, F. Ruess, M. Suppa, D. Burschka, Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue, IEEE Robot. Autom. Mag. 19 (3) (2012) 46–56.
[5] M. Erdelj, E. Natalizio, K.R. Chowdhury, I.F. Akyildiz, Help from the sky: Leveraging UAVs for disaster management, IEEE Pervasive Comput. 16 (1) (2017) 24–32.
[6] J. Zhang, S. Xu, Y. Zhao, J. Sun, S. Xu, X. Zhang, Aerial orthoimage generation for UAV remote sensing, Inf. Fusion 89 (2023) 91–120.
[7] Z. Ning, P. Dong, M. Wen, X. Wang, L. Guo, R.Y. Kwok, H.V. Poor, 5G-enabled UAV-to-community offloading: Joint trajectory design and task scheduling, IEEE J. Sel. Areas Commun. 39 (11) (2021) 3306–3320.
[8] E. Besada-Portas, L. De La Torre, A. Moreno, J.L. Risco-Martín, On the performance comparison of multi-objective evolutionary UAV path planners, Inform. Sci. 238 (2013) 111–125.
[9] G. Skorobogatov, C. Barrado, E. Salamí, Multiple UAV systems: A survey, Unmanned Syst. 8 (02) (2020) 149–169.
[10] Y. Niu, X. Yan, Y. Wang, Y. Niu, Three-dimensional collaborative path planning for multiple UCAVs based on improved artificial ecosystem optimizer and reinforcement learning, Knowl.-Based Syst. 276 (2023) 110782.
[11] J. Chen, F. Ling, Y. Zhang, T. You, Y. Liu, X. Du, Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system, Swarm Evol. Comput. 69 (2022) 101005.
[12] X. Yu, N. Jiang, X. Wang, M. Li, A hybrid algorithm based on grey wolf optimizer and differential evolution for UAV path planning, Expert Syst. Appl. 215 (2023) 119327.
[13] X. Zhang, S. Xia, X. 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.
[14] T. Zhou, P. He, B. Niu, G. Yue, H. Wang, A novel competitive constrained dual-archive dual-stage evolutionary algorithm for constrained multiobjective optimization, Swarm Evol. Comput. 83 (2023) 101417.
[15] J. Wu, D. Li, J. Shi, X. Li, L. Gao, L. Yu, G. Han, J. Wu, et al., An adaptive conversion speed Q-learning algorithm for search and rescue UAV path planning in unknown environments, IEEE Trans. Veh. Technol. 72 (12) (2023) 15391–15404.
[16] Y. Zhou, X. Kong, K.-P. Lin, L. Liu, Novel task decomposed multi-agent twin delayed deep deterministic policy gradient algorithm for multi-UAV autonomous path planning, Knowl.-Based Syst. 287 (2024) 111462.
[17] Y. Zeng, S. Chen, Y. Cui, J. Yang, Y. Fu, Joint resource allocation and trajectory optimization in UAV–enabled wirelessly–powered MEC for large area, IEEE Internet Things J. 10 (17) (2023) 15705–15722.
[18] E.L. Kulida, V.G. Lebedev, Genetic algorithm for generating trajectories of specified length for the onboard flight path safety system, Procedia Comput. Sci. 112 (2017) 1015–1022.
[19] L.E. Kavraki, P. Svestka, J.-C. Latombe, M.H. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces, IEEE Trans. Robot. Autom. 12 (4) (1996) 566–580.
[20] S. Aggarwal, N. Kumar, Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges, Comput. Commun. 149 (2020) 270–299.
[21] J. Fan, X. Chen, Y. Wang, X. Chen, UAV trajectory planning in cluttered environments based on PF-RRT* algorithm with goal-biased strategy, Eng. Appl. Artif. Intell. 114 (2022) 105182.
[22] S. Bayili, F. Polat, Limited-damage A*: A path search algorithm that considers damage as a feasibility criterion, Knowl.-Based Syst. 24 (4) (2011) 501–512.
[23] J. Wang, Y. Li, R. Li, H. Chen, K. Chu, Trajectory planning for UAV navigation in dynamic environments with matrix alignment Dijkstra, Soft Comput. 26 (22) (2022) 12599–12610.
[24] L. Wirth, P. Oettershagen, J. Ambühl, R. Siegwart, Meteorological path planning using dynamic programming for a solar-powered UAV, in: 2015 IEEE Aerospace Conference, IEEE, 2015, pp. 1–11.
[25] X. Zhang, F. Zhang, Z. Tang, X. Chen, A MILP model on coordinated coverage path planning system for UAV-ship hybrid team scheduling software, J. Syst. Softw. 206 (2023) 111854.
[26] A. Puente-Castro, D. Rivero, A. Pazos, E. Fernandez-Blanco, A review of artificial intelligence applied to path planning in UAV swarms, Neural Comput. Appl. (2022) 1–18.
[27] Z. Pan, L. Wang, C. Dong, J.-f. Chen, A knowledge-guided end-to-end optimization framework based on reinforcement learning for flow shop scheduling, IEEE Trans. Ind. Inform. 20 (2) (2024) 1853–1861.
[28] F. Zhao, S. Di, L. Wang, A hyperheuristic with Q-learning for the multiobjective energy-efficient distributed blocking flow shop scheduling problem, IEEE Trans. Cybern. 53 (5) (2023) 3337–3350.
[29] J.-J. Wang, L. Wang, A cooperative memetic algorithm with learning-based agent for energy-aware distributed hybrid flow-shop scheduling, IEEE Trans. Evol. Comput. 26 (3) (2021) 461–475.
[30] F. Zhao, G. Zhou, L. Wang, A cooperative scatter search with reinforcement learning mechanism for the distributed permutation flowshop scheduling problem with sequence-dependent setup times, IEEE Trans. Syst. Man Cybern. Syst. 53 (8) (2023) 4899–4911.
[31] Z. Lin, K. Gao, N. Wu, P.N. Suganthan, Scheduling eight-phase urban traffic light problems via ensemble meta-heuristics and Q-learning based local search, IEEE Trans. Intell. Transp. Syst. 24 (12) (2023) 14415–14426.
[32] Y. Ren, K. Gao, Y. Fu, H. Sang, D. Li, Z. Luo, A novel Q-learning based variable neighborhood iterative search algorithm for solving disassembly line scheduling problems, Swarm Evol. Comput. 80 (2023) 101338.
[33] Y. Wan, Y. Zhong, A. Ma, L. 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.
[34] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, C.P. Chen, Multiobjective overtaking maneuver planning for autonomous ground vehicles, IEEE Trans. Cybern. 51 (8) (2020) 4035–4049.
[35] C. Peng, X. Huang, Y. Wu, J. Kang, Constrained multi-objective optimization for UAV-enabled mobile edge computing: Offloading optimization and path planning, IEEE Wireless Commun. Lett. 11 (4) (2022) 861–865.
[36] J.-S. Dong, Q.-K. Pan, Z.-H. Miao, H.-Y. Sang, L. Gao, An effective multi-objective evolutionary algorithm for multiple spraying robots task assignment problem, Swarm Evol. Comput. 87 (2024) 101558.
[37] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, Multi-objective trajectory optimization of space manoeuvre vehicle using adaptive differential evolution and modified game theory, Acta Astronaut. 136 (2017) 273–280.
[38] Y. Zhao, Z. Zheng, X. Zhang, Y. Liu, Q learning algorithm based UAV path learning and obstacle avoidence approach, in: 2017 36th Chinese Control Conference, CCC, IEEE, 2017, pp. 3397–3402.
[39] W. Zhao, Z. Fang, Z. Yang, Four-dimensional trajectory generation for UAVs based on multi-agent Q learning, J. Navig. 73 (4) (2020) 874–891.
[40] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, Solving constrained trajectory planning problems using biased particle swarm optimization, IEEE Trans. Aerosp. Electron. Syst. 57 (3) (2021) 1685–1701.
[41] L. Xu, X. Cao, W. Du, Y. Li, Cooperative path planning optimization for multiple UAVs with communication constraints, Knowl.-Based Syst. 260 (2023) 110164.
[42] A. Sathyan, N.D. Ernest, K. Cohen, An efficient genetic fuzzy approach to UAV swarm routing, Unmanned Syst. 4 (02) (2016) 117–127.
[43] R. Chai, A. Savvaris, A. Tsourdos, Y. Xia, S. Chai, Solving multiobjective constrained trajectory optimization problem by an extended evolutionary algorithm, IEEE Trans. Cybern. 50 (4) (2018) 1630–1643.
[44] Y. Tian, T. Zhang, J. Xiao, X. Zhang, Y. Jin, A coevolutionary framework for constrained multiobjective optimization problems, IEEE Trans. Evol. Comput. 25 (1) (2020) 102–116.
[45] X. Dai, B. Duo, X. Yuan, W. Tang, Energy-efficient UAV communications: A generalized propulsion energy consumption model, IEEE Wireless Commun. Lett. 11 (10) (2022) 2150–2154.
[46] A. Thibbotuwawa, P. Nielsen, B. Zbigniew, G. Bocewicz, Energy consumption in unmanned aerial vehicles: A review of energy consumption models and their relation to the UAV routing, in: Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology–ISAT 2018: Part II, Springer, 2019, pp. 173–184.

[47] H. Zhang, X. Gan, S. Li, Z. Chen, UAV safe route planning based on PSO-BAS algorithm, J. Syst. Eng. Electron. 33 (5) (2022) 1151–1160.
[48] J. Sun, W. Fang, X. Wu, V. Palade, W. Xu, Quantum-behaved particle swarm optimization: analysis of individual particle behavior and parameter selection, Evol. Comput. 20 (3) (2012) 349–393.
[49] C.C. Coello, M.S. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), Vol. 2, IEEE, 2002, pp. 1051–1056.
[50] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, in: Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18–20, 2000 Proceedings 6, Springer, 2000, pp. 849–858.
[51] F. Ming, W. Gong, L. Wang, A two-stage evolutionary algorithm with balanced convergence and diversity for many-objective optimization, IEEE Trans. Syst. Man Cybern. Syst. 52 (10) (2022) 6222–6234.