

Multi-unmanned aerial vehicle swarm formation control using hybrid strategy

Zain Anwar Ali  and Han Zhangang

Transactions of the Institute of
Measurement and Control
2021, Vol. 43(12) 2689–2701
© The Author(s) 2021

Article reuse guidelines:

sagepub.com/journals-permissions

DOI: 10.1177/01423312211003807

journals.sagepub.com/home/tim



Abstract

This study proposes a novel hybrid strategy for formation control of a swarm of multiple unmanned aerial vehicles (UAVs). To enhance the fitness function of the formation, this research offers a three-dimensional formation control for a swarm using particle swarm optimization (PSO) with Cauchy mutant (CM) operators. We use CM operators to enhance the PSO algorithm by examining the varying fitness levels of the local and global optimal solutions for UAV formation control. We establish the terrain and the fixed-wing UAV model. Furthermore, it also models different control parameters of the UAV as well. The enhanced hybrid algorithm not only quickens the convergence rate but also improves the solution optimality. Lastly, we carry out the simulations for the multi-UAV swarm under terrain and radar threats and the simulation results prove that the hybrid method is effective and gives better fitness function.

Keywords

Fixed-wing UAV, swarm formation, Cauchy mutant (CM), particle swarm optimization, leader-follower formation

Introduction

The unmanned aerial vehicle (UAV) is a plane that does not have a person on board. UAV has extensive applications for both civilian and military missions, for instance, product deliveries, search and rescue, surveillance, and commercial purposes (Shakhathreh et al., 2019). As opposed to a single UAV, a multi-UAV swarm has distinct benefits of better efficiency, lower cost, and bigger coverage than a single vehicle.

Multi-UAV formation control has drawn the significant interest of the academic and scientific community (Chen et al., 2014b). In earlier studies, scientists have modeled the path planning for a multi-UAV as an optimization problem. These models intend to find the most optimal route from origin to target under different constraints and circumstances. The objective of these models is to achieve the least flight route, the least flight time, collision avoidance, obstacle avoidance, and handling communication delays, and so forth (Cetin and Yilmaz, 2016; Duan and Shen, 2017; Duan et al., 2016). Optimizing the path planning is crucial in enhancing the autonomy and intelligence of unmanned aerial systems (Jabbarpour et al., 2014).

In recent years, researchers have presented different techniques for the path planning of autonomous multi-UAV formations. Algorithms based on graphs like Voronoi diagram (Pehlivanoglu, 2012), probabilistic roadmap technique (Marble and Bekris, 2013), rapidly exploring random trees method (Kothari and Postlethwaite, 2013), are some of the simple methods for path planning. However, these algorithms rarely consider UAV dynamic and environmental constraints and therefore are not reliable for real-world scenarios. Reza Olfati-Saber (2006) presented one of the more effective algorithms. He proposed a theoretic outline for the design and

analysis of the distributed flocking algorithm for the multiple UAV formation. Another efficient technique for path planning is an algorithm based on the potential field. Some examples are interfered fluid dynamical system technique (Yao et al., 2016) and artificial potential field method (Chen et al., 2014a; Sun et al., 2017).

To create a flightworthy route, such techniques must form global coordination amongst the attractive and repulsive field. Consequently, the algorithm can easily fall into the local optimum, and sometimes they are not able to generate a viable path when the obstacles or target is near. With the progress in swarm intelligence, the bio-inspired algorithms have made much progress (Duan and Luo, 2015; Rajput and Kumari, 2017). These algorithms can find the best result more efficiently and robustly. Many scientists use these algorithms for the path planning multi-UAV formations. The most frequently used bio-inspired algorithms are the artificial bee colony (ABC) method (Yaghoobi and Esmaili, 2017), the ant colony optimization (ACO) technique (Stodola and Mazal, 2016), the genetic algorithm (GA) method (Roberge et al., 2018) and particle swarm optimization (PSO) method (Das et al., 2016; Xia et al., 2019).

School of Systems Science, Beijing Normal University, China

Corresponding authors:

Zain Anwar Ali, School of Systems Science, Beijing Normal University, China.

Email: zainanwar86@hotmail.com

Han Zhangang, School of Systems Science, Beijing Normal University, China.

Email: zhhan@bnu.edu.cn

The two critical factors affecting the path planning are convergence speed and solution optimality. For real-world applications, researchers always prefer the routes with quicker convergence speed and the most optimal outcome. Though some swarm algorithms have good performance, the solution optimality and convergence speed are not feasible enough for the real-world flying applications. Most existing research focus on the path planning of a single UAV, which may not be good enough for a swarm of UAVs.

Swarm intelligence has the advantage of simple implementation and can achieve the best global result. Researchers use it for multi-UAV formation control and further optimization problems. In addition to the applications in multi-UAV formation control, several academics have also concentrated on the theoretical studies and enhancements in swarm intelligence. Researchers have carefully examined the aspects influencing the outcomes of PSO and concluded that the four key factors in enhancing the effectiveness of PSO are topology structure, parameter selection, hybrid methods, and swarm initialization.

One main issue with the traditional PSO is that it could easily fall into the local optimum in many optimization problems. Several researchers have tried to resolve this issue. One study (Tian and Shi, 2018) added a mutation mechanism and chaotic maps into a classic PSO to avoid falling into local optimum and premature solutions. With the help of prior research, quicker convergence and more optimal solutions are the primary drivers of this research. Hence, we propose a new hybrid algorithm based on PSO and Cauchy mutant (CM) operators. The CM operators ensure the best solution in less time than a standard PSO algorithm. Cauchy Landmark operators reduce the possibility of immature convergence by dealing with each ant separately. All distinct ants will progressively reach for the global optimal solution.

The major offerings of this study are as follows. This research introduces a novel hybrid technique not present in the

simulation scenarios and their results. Lastly, Section 7 concludes the entire research and introduces some future research ideas.

Problem formulation

Some vital aspects such as the atmosphere, UAV protection, and the cost of each path play a significant role in the path planning of each UAV in the formation. Overall, the mission area primarily comprises the territory, radars, and obstacles. The mission goal should take into account all the essential ecological features. It should also assess how these features affect the formation. Afterwards, this paper models the path planning as an optimization problem and solves it using the suggested hybrid algorithm. This section now introduces the environmental parameters and mission objective below.

Flying range

The objective is to discover the best route to the target under difficult terrain and environmental parameters. x, y, z represent the coordinates of a point in the three-dimensional (3D) atmosphere. The flying range of the mission area is rewritten as (Huang and Fei, 2018)

$$\{(x, y, z) | x_{min} \leq x \leq x_{max}, y_{min} \leq y \leq y_{max}, z_{min} \leq z \leq z_{max}\} \quad (1)$$

where $x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}$ represent the flying constraints correspondingly.

Terrain model

A frequent function of terrain models is to create a datum, peak, and plateau terrain, which are given below

$$\begin{cases} Z_1(x, y) = \sin(y + a) + b \cdot \sin(x) + c \cdot \cos(d\sqrt{x^2 + y^2}) + e \cdot \cos(x) + f \cdot \sin(f\sqrt{x^2 + y^2}) + g \cdot \cos(y) \\ Z_2(x, y) = \sum_{k=1}^{k=N} h_k \exp \left[-\left(\frac{x - x_{0k}}{x_{sk}} \right)^2 - \left(\frac{y - y_{0k}}{y_{sk}} \right)^2 \right] + Z_0 \\ Z(x, y) = \max(Z_1, Z_2) \end{cases} \quad (2)$$

existing literature by combining the classic PSO with the CM Operators. It increases the complexity of the algorithm but also enhances the fitness. This study implements the designed hybrid algorithm effectively for multi-UAV swarm formation under the radar, mountainous terrain, and obstacle avoidance parameters. Finally, the fitness of the multi-UAV formation is evaluated in detail and the results are better than similar techniques, like the one described by Zhou et al. (2016).

The divisions of this study are: Section 2 designs the model for flying space and the mountainous terrain. It also deals with the various parameters including radar and collision cost. Section 3 presents the preliminaries of UAV Formation with communication delay and discusses the communication graph concept of UAVs. Section 4 presents the fixed-UAV model and the leader-follower formation model. Section 5 discusses the classic PSO algorithm and its constraints. It also deals with Cauchy mutant operators and presents the pseudo-code of the proposed algorithm. Section 6 offers different

In the datum terrain function Z_1 , (x, y) are the Cartesian coordinates, and (a, b, c, d, e, f, g) are constant numbers. The constants vary according to changes in height.

In the peak terrain function Z_2 , (x_{0k}, y_{0k}) is the center of peak k , h_k is the height of the k th peak. (x_{sk}, y_{sk}) is the drop in (x, y) coordinates of the k th peak, Z_0 is the initial height, and N is the total amount of peaks.

The last plateau terrain expression Z is a maximum value of Z_1 and Z_2 .

Objective function design

Finding optimal swarm paths is a complex multi-faceted problem. Taking the distance, obstacle avoidance, and terrain and radar parameters into account, the evaluation function for a swarm is as follows

$$f = f_L + f_T + f_R + f_C + f_H \quad (3)$$

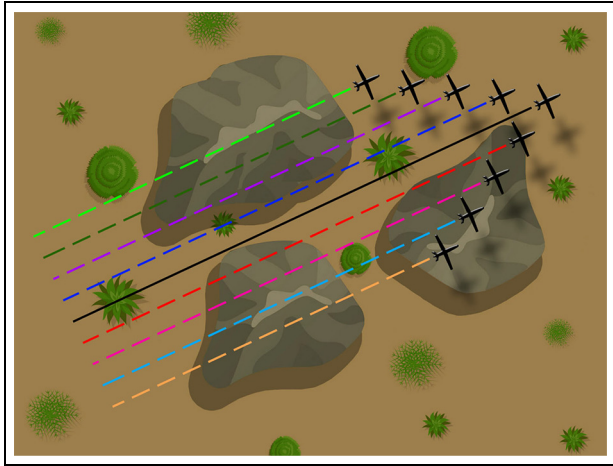


Figure 1. The mission area including mountains.

whereas f_L represents the route length cost, f_T represents the mountainous environment cost. While f_R is the radar threat cost, f_C is the collision cost and f_H is the altitude changing cost.

(1) Minimal flying path length cost

UAV path planning must take into consideration the finite quantity of fuel. Therefore, the path must be as short as possible. Supposing that a whole route consists of n legs, the minimal flying path length cost is

$$f_L = \sum_{k=1}^n l_k \tag{4}$$

where l_k denotes the length path leg.

(2) Mountain terrain cost

The space that separates the UAVs and the mountains determine the terrain cost. The nearer the UAVs are to the mountain, the higher the mountain terrain cost will be. The mountain terrain cost is

$$f_T = \begin{cases} \frac{K}{avgR}, path \ not \ through \ the \ mountains \\ \varepsilon, path \ through \ the \ mountains \end{cases} \tag{5}$$

$$avgR = \frac{\sum_{k=1}^n r_k}{n} \tag{6}$$

where K is a number that relates to the real flying range, ε is the penalty constant, n is total legs, r_k is the distance between the path leg and the mountain terrain.

This paper uses the mountainous terrain for the first scenario 1 of our simulation. Figure 1 shows the mission area with mountainous terrain.

(3) The radar threat cost

The enemy might detect or attack the UAV if it goes into the radar detection range. The longer the space amid the UAVs

and the radar, the lesser the chance that the UAVs are detected. The radar cost is (Yang et al., 2015)

$$f_R = \sum_{k=2}^{N_w} \sum_{l=1}^n B_{k,l} \text{with } B_{k,l} = \begin{cases} \left(\frac{\delta}{D_{k,l}}\right)^4, \text{ if } D_{k,l} \leq R_t \\ 0, \text{ otherwise} \end{cases} \tag{7}$$

whereas

$$D_{k,l} = \sqrt{(x_{k,l} - x_t^r)^2 + (y_{k,l} - y_t^r)^2 + (z_{k,l} - z_t^r)^2} \tag{8}$$

$D_{k,l}$ represents the distance between the path point $(x_{k,l}, y_{k,l}, z_{k,l})$ and the radar center (x_t^r, y_t^r, z_t^r) . While δ represents the radar intensity, R_t represents the threat radius, N_w is the number of total path waypoints, and n is the number of radars.

(4) The collision cost

In multiple UAVs swarm, each UAV encounters obstacles and other UAVs of the swarm. Therefore, the formation has to take into account two problems at the same time; formation control and collision avoidance. It can only maintain the formation if it is at enough distance from both the obstacles and other UAVs.

To accomplish collision avoidance, Wang et al. (2007) uses an indexing technique. This technique tags every UAV and gives them a higher or lower index. A lower indexed UAV constructs a virtual obstacle around the higher indexed UAV and attempts to dodge it.

To avoid any collision, lower indexed UAVs must respond quickly when nearby higher indexed UAVs or other obstacles come within dangerous range. Therefore, a cost function of collision avoidance is

$$f_C = P_c \times \sum_{l=1}^{num_c} d_{kl} \tag{9}$$

$$d_{kl} = \begin{cases} 1, d_{kl} \leq d_{safe} \\ 0, d_{kl} > d_{safe} \end{cases} \tag{10}$$

whereas P_c represents the penalty due to collision, and num_c is the amount of collisions that UAV_k must avoid. While d_{kl} is the space between UAV_k and the l th collision center, d_{safe} is the space that UAVs are required to maintain among themselves to avoid a collision.

(5) Altitude changing cost

UAV has to lower or raise its altitude to avoid colliding into a mountain, UAVs, and other obstacles. The UAV also has to change its altitude to maintain the formation. However, this move consumes fuel. Moreover, low temperature negatively affects engine performance; repeated movements may be a danger for the safety of the aircraft. Hence, the algorithm must prevent regular variations in height. The height varying cost is given as

$$f_H = \sqrt{\frac{1}{n} \sum_{k=0}^n \left(z_k - \frac{1}{n+1} \sum_{k_1=0}^n z_{k_1} \right)^2} \tag{11}$$

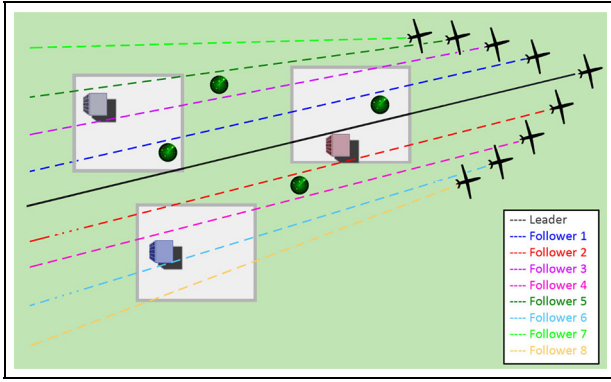


Figure 2. Mission environment with obstacles and radar.

Figure 2 shows the mission area with obstacles (tall buildings) and radars that is used for scenario 2. Collision avoidance and altitude changing are crucial in this scenario to reach the target without any accident.

Constraints of the optimization model

The constraints for the proposed optimization model can be written as

$$\begin{cases} l_{min} \leq l_i \leq l_{max} & i = 1, 2, \dots, n \\ \psi_i \leq \psi_{max} & i = 1, 2, \dots, n \\ \vartheta_i \leq \vartheta_{max} & i = 1, 2, \dots, n \\ H_i < H_{max} & i = 1, 2, \dots, n \end{cases} \quad (12)$$

Whereas l represents the flight length, l_{min} and l_{max} represent the minimum and maximum allowed flight length, ψ is the turning angle, ψ_{max} is the maximum allowed turning angle, ϑ is the climb/dive angle and ϑ_{max} is the maximum climb/dive angle. Similarly, H is the altitude and H_{max} is the maximum allowed altitude.

Preliminaries of UAV formation with communication delay

During the flight of multiple UAV formations, formation members ensure the consistency of the formation configuration through mutual information transmission and sharing. In fact, in the process of the formation of members' information exchange, the communication delay is inevitable. This situation has a certain impact on the stability of the entire system, so this section has some practical significance to study the consistency problem of the UAV formation system with communication delay.

Communication graph concept of UAVs

Suppose $G(V, E, A)$ represents a directed graph of communication topology when several drones are flying in formation, where $V = \{V_1, V_2, \dots, V_k\}$ is the vertex set, E is the edge set, and A is the weight adjacency matrix.

The edge of a directed graph can be expressed as $e_{pq} = (V_p, V_q)$, where V_p is expressed as the tail of this edge, and V_q is expressed as the head of this edge. The weight adjacency matrix $A = [a_{pq}]$, where the matrix elements represent the adjacency weight; $a_{pq} > 0$ means that communication node p can get the information of node q , otherwise $a_{pq} = 0$.

Whereas the diagonal matrix $D = \text{diag}\{d_p, p=1, 2, \dots, k\}$ has the elements of the i^{th} row of matrix A added to get d_p , and the Laplace matrix of graph G is defined as $L = D - A$, where G is an undirected graph, that is, $a_{pq} = a_{qp}$. While L is a symmetric positive semi-definite matrix. If G is undirected and any two nodes in it can be connected by edges, then G is called an undirected connected graph. From the basic knowledge of graph theory, we know that the matrix L can be diagonalized to the minimum value of diagonal matrix Γ , that is, 0 that satisfies the following

$$0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_{max} \quad (13)$$

Whereas $\lambda_1 \sim \lambda_{max}$ are the eigenvalues of Laplace matrix L .

Suppose that the communication topology of a formation system consisting of n drones is an undirected connected graph. The dynamic system model of the UAV is approximately defined by the following second-order model

$$\begin{cases} \dot{x}_p(t) = v_p(t) \\ \dot{v}_p(t) = u_p(t) \end{cases} p = 1, 2, \dots, k \quad (14)$$

Whereas x_p is the position status of the formation member p , v_p is the speed status of formation member p , and u_p is the control input of formation member p . To make sure that the members of the formation are consistent with their expected movement status, the following control protocol is adopted

$$\begin{aligned} u_p(t) = & \sum a_{pq} [k_1(x_q(t-\tau) - x_p(t-\tau) + r_{pq}) + k_2(v_q(t-\tau) - v_p(t-\tau))] \\ & + \sum^{k_3} h_p(v_s - v_p) \end{aligned} \quad (15)$$

Whereas k_1, k_2, k_3 are the control gains of the system, τ is the time delay for information to transfer from formation members q to p . While r_{pq} is the expected relative position of the formation members p and q ; v_s is the expected speed of the formation, and h_p is the ability index to obtain the expected speed information of the formation. The matrix of the above control protocol expressed as

$$\begin{aligned} U(t) = & -k_1(LX(t-\tau) - \text{diag}(AR)) - k_2LV(t-\tau) \\ & - k_3H(V(t) - v_s \otimes 1_{n \times 1}) \end{aligned} \quad (16)$$

Where R is the $n \times n$ matrix formed by r_{ij} , and $\text{diag}(AR)$ is a vector of $A \times R$ diagonal elements; H is the $n \times n$ diagonal matrix formed by h_i ; x is a vector composed of the position status of formation members, $X(t) = [x_1(t), x_2(t), \dots, x_k]^T$; $V(t)$ is the vector of the formation members' speed states, $V(t) = [v_1(t), v_2(t), \dots, v_k]^T$; and $U(t)$ is the vector formed by the control input of formation members, $U(t) = [u_1(t), u_2(t), \dots, u_k]^T$. Then, the state equation of the system can be written as

$$\begin{aligned} \begin{bmatrix} \dot{X} \\ \dot{V} \end{bmatrix} &= \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \otimes \mathbf{I}_k - \begin{bmatrix} 0 & 0 \\ 0 & k_3 \end{bmatrix} \otimes \mathbf{H} \right) \begin{bmatrix} X \\ V \end{bmatrix} \\ &- \left(\begin{bmatrix} 0 & 0 \\ k_1 & k_2 \end{bmatrix} \otimes \mathbf{L} \right) \begin{bmatrix} X(t-\tau) \\ V(t-\tau) \end{bmatrix} + \begin{bmatrix} 0 \\ k_1 \end{bmatrix} \otimes \text{diag}(\mathbf{AR}) + \\ &\begin{bmatrix} 0 \\ k_3 v_s \end{bmatrix} \otimes (\mathbf{H} \times \mathbf{I}_{k \times 1}) \end{aligned} \quad (17)$$

It can be seen from the state equations of the above system that the goal of the control protocol is

$$\begin{cases} \lim_{t \rightarrow \infty} \|x_p(t) - x_q(t)\| \rightarrow r_{pq} \\ \lim_{t \rightarrow \infty} \|v_p(t) - v_q(t)\| \rightarrow 0 \\ \lim_{t \rightarrow \infty} v_p(t) = \lim_{t \rightarrow \infty} v_q(t) \rightarrow v_s \end{cases} \quad (18)$$

That is, the relative positions of formation members p and q approach the expected relative position r_{pq} , and the speeds of p and q both approach the expected speed v_s . The above is the mathematical description of the consistency problem of a UAV formation system.

Formation modelling and control

Fixed-wing UAV model

This study primarily concentrates on the control of the multi-UAV swarm. We use the point-mass system to model our fixed-wing UAVs (Huang et al., 2016; Wang and Xin, 2012). Take a swarm of UAVs into consideration using the model given below

$$\dot{x}_k = V_k \cos \gamma_k \cos \chi_k \quad (19)$$

$$\dot{y}_k = V_k \cos \gamma_k \sin \chi_k \quad (20)$$

$$\dot{h}_k = V_k \sin \gamma_k \quad (21)$$

$$V_k = \frac{T_k - D_k}{m_k} - g \sin \gamma_k \quad (22)$$

$$\dot{\gamma}_k = \frac{L_k \cos \varphi_k - m_k g \cos \gamma_k}{m_k V_k} \quad (23)$$

$$\dot{\chi}_k = \frac{L_k \sin \varphi_k}{m_k V_k \cos \gamma_k} \quad (24)$$

where $k = 1, \dots, M$, M is the number of UAVs, x_k and y_k are the forward and lateral displacement, respectively. h_k is the height, V_k is the ground velocity. γ_k is the flying path angle, φ_k is the banking angle, and χ_k is the heading angle. T_k is the thrust, D_k is the drag and L_k is the lift. m_k and g are the mass and the gravitational acceleration, respectively.

Using feedback linearization, the nonlinear model can be linearized as

$$\begin{cases} \ddot{x}_k = u_{x_k} \\ \ddot{y}_k = u_{y_k} \\ \ddot{h}_k = u_{h_k} \end{cases} \quad (25)$$

where u_{x_k} , u_{y_k} , and u_{h_k} are simulated control inputs linked with the UAV movements and height changes. The real control inputs are banking angle, lift, and thrust, which are written as

$$\varphi_k = \tan^{-1} \left(\frac{u_{y_k} \cos \chi_k - u_{x_k} \sin \chi_k}{(u_{h_k} + g) \cos \gamma_k - (u_{x_k} \cos \chi_k + u_{y_k} \sin \chi_k) \sin \gamma_k} \right) \quad (26)$$

$$L_k = \frac{m_k (u_{h_k} + g) \cos \gamma_k - (u_{x_k} \cos \chi_k + u_{y_k} \sin \chi_k) \sin \gamma_k}{\cos \varphi_k} \quad (27)$$

$$T_k = m_k [(u_{h_k} + g) \sin \gamma_k + (u_{x_k} \cos \chi_k + u_{y_k} \sin \chi_k) \cos \gamma_k] + D_k \quad (28)$$

Additionally, V_k , χ_k , and h_k must meet the following dynamic parameters

$$\begin{cases} V_{min} \leq V_k \leq V_{max} \\ |\chi_k| \leq n_{max} g / V_k \\ \lambda_{min} \leq h_k \leq \lambda_{max} \end{cases} \quad (29)$$

where V_{min} and V_{max} are the minimum and maximum velocity, n_{max} is the maximum lateral overload, λ_{min} and λ_{max} are the minimum and maximum climbing speed.

Leader-follower formation control

We use the point mass model for the formation control of the multi-UAV swarm. Every UAV flies at a certain height, parallel to the 3D mission area. The model used for the formation is given below

$$\begin{cases} \dot{x} = V_F \cos \psi_E \cos \mu_E - \dot{\mu}_L \cdot z + \dot{\psi}_L \cdot y - V_L \\ \dot{y} = V_F \sin \psi_E \cos \mu_E - \dot{\psi}_L \cdot x \\ \dot{z} = V_F \sin \mu_E - \dot{\mu}_L \cdot x \end{cases} \quad (30)$$

$$\begin{cases} \mu_E = \mu_L^f - \mu_F^f \\ \psi_E = \psi_L^f - \psi_F^f \end{cases} \quad (31)$$

whereas (x, y, z) are the current coordinates of the UAV, V_F and V_L are the angular and lateral velocity and ψ_E and μ_E are the heading angles in the (x, y, z) plane. Figure 3 shows the desired leader-follower configuration.

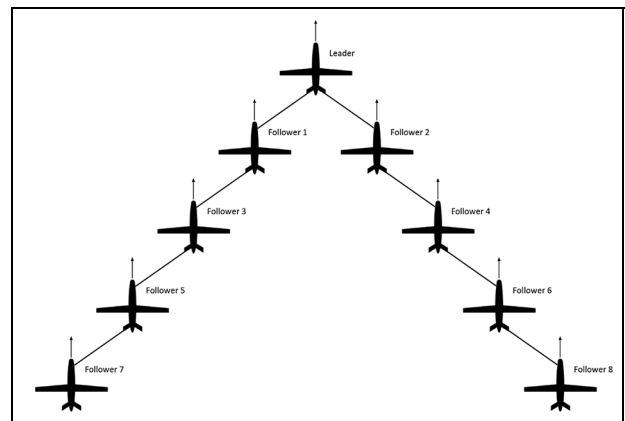


Figure 3. Leader-follower configuration.

In a usual multi-UAV swarm formation, the followers pursue the path of the leader, keeping other UAVs as references to maintain the formation. In a big formation, spaces between UAVs must remain constant (He et al., 2018). In our case, this is only true for scenario 1 with no dynamic obstacles. Since scenario 2 contains dynamic threats, maintaining a tight formation is difficult, hence there will be slight variations in distances between UAVs. This paper adopts leader-follower formation control; that is, each follower plans its path according to the leader, while the height remains the same for every UAV. The leader UAV is always ahead of every follower.

Our model uses a simulated leader instead of a real leader so that followers can fine-tune their velocity and direction

$$\begin{cases} P_{(k,best)} = [P_{(k,1,best)}, \dots, P_{(k,D,best)}]^T = [(P^x, P^y, P^z)_{(k,1,best)}, \dots, (P^x, P^y, P^z)_{(k,D,best)}]^T \\ P_{best}^G = [P_{1,best}, \dots, P_{D,best}]^T = [(P^x, P^y, P^z)_{(1,best)}, \dots, (P^x, P^y, P^z)_{(D,best)}]^T \end{cases} \quad (35)$$

according to the simulated leader. The main benefit of the simulated leader is that a real leader can become dysfunctional, while the simulated leader remains unharmed. The simulated leader offers a firm and effective formation control (Xuan-Mung and Hong, 2019).

Hybrid strategy for the formation control

PSO

PSO is a computational method based on the analysis of natural behaviors of fish schooling and bird flocking. The PSO technique initializes with random solutions. It then repeatedly updates the position and velocity to obtain a global best solution (Cheng and Jin, 2015; Liu et al., 2016). Every particle dynamically updates its searching direction using its preceding outcome, the individual best outcome, and the global best outcome. Its updated position primarily relies on its earlier position and the present velocity. Given the newly acquired particle position is superior to its earlier position, the new position becomes the individual best position. Given the new particle position is superior to all other positions, it becomes the global best position.

Consider path planning in a 3D space and assume D is the waypoint of every particle, then the position and velocity vector P_k and V_k , respectively, for the k th particle are

$$\begin{aligned} P_k &= [P_{(k,1)}, \dots, P_{(k,D)}]^T \\ &= [(P^x, P^y, P^z)_{(k,1)}, \dots, (P^x, P^y, P^z)_{(k,D)}]^T \end{aligned} \quad (32)$$

$$\begin{aligned} V_k &= [V_{(k,1)}, \dots, V_{(k,D)}]^T \\ &= [(V^x, V^y, V^z)_{(k,1)}, \dots, (V^x, V^y, V^z)_{(k,D)}]^T \end{aligned} \quad (33)$$

where, $l \in 1, \dots, D$; $(P^x, P^y, P^z)_{(k,l)}$, $(V^x, V^y, V^z)_{(k,l)}$ represent the l th waypoint's position and velocity of the k th particle in 3D.

If total particles are S , the swarm is given as

$$[(P_1, V_1), (P_2, V_2), \dots, (P_S, V_S)] \quad (34)$$

For PSO algorithm having S particles, there are the S individual optimal positions and one global optimal position, which is

whereas $k \in 1, \dots, S$ is the particle number.

Now, for the multi-objective swarm function

$$P_{(k,best)}(t+1) = \begin{cases} P_{(k,best)}(t); \text{ iff } (P_{(k,best)}(t)) \leq f(P_k(t+1)) \\ P_k(t+1); \text{ iff } (P_{(k,best)}(t)) > f(P_k(t+1)) \end{cases} \quad (36)$$

$$P_{best}^G \in (P_{(k,1,best)}, \dots, P_{(k,D,best)}) \quad (37)$$

$$f(P_{best}^G(t)) = \min[f(P_{best,1}(t)), \dots, f(P_{best,D}(t))] \quad (38)$$

Each particle in the swamp updates according to the following equations

$$\begin{cases} P_{k,l}(t+1) = P_{k,l}(t) + V_{k,l}(t+1) \\ V_{k,l}(t+1) = w \cdot V_{(k,l)}(t) + a_1 \cdot r_1 \cdot (P_{(k,l,best)}(t) - P_{(k,l)}(t)) + a_2 \cdot r_2 \cdot (P_{(G,l,best)}^G(t) - P_{(k,l)}(t)) \end{cases} \quad (39)$$

whereas a_1 and a_2 are accelerating coefficients, r_1 , r_2 are any two random numbers from 0 to 1.

We can improve PSO by modifying the inertia weight w . It stabilizes the local and global outcomes through the search procedure. By increasing the inertia weight, we can enhance the global search. To improve the local search, decrease the inertia weight, which is

$$w = \frac{(w_{max} - w_{min})t}{T} \quad (40)$$

where t is the current iteration, T is the maximum iterations, w_{max} and w_{min} represent the maximum and minimum value of w , respectively.

Cauchy mutant operators

If x meets the condition given in the following equation, then the below function becomes Cauchy distribution, which is

$$f(x; x_0, \gamma) = \frac{1}{\pi} \left[\frac{\gamma}{(x - x_0)^2 + \gamma^2} \right], x \in (-\infty, +\infty) \quad (41)$$

whereas x_0 is the highest value of the function and γ is the width related to half of x_0 . When γ and x_0 are 1 and 0, respectively, x meets the condition of the probability density function, and equation (41) becomes

$$F(x; 0, 1) = \frac{1}{\pi} \arctan(x) + \frac{1}{2}, x \in (-\infty, +\infty) \quad (42)$$

(1) Map operators and compass operators

For the classic PSO algorithm, we use these operators to find the global best outcomes during the search problem. Thus, the best outcomes determine the particle's velocity and position. Cauchy Mutant (CM) operator c_1 can increase the searching area. It can also avoid falling into the local optimum. The CM operator c_1 written as

$$\begin{cases} \frac{1}{\pi} \arctan(c_1) + \frac{1}{2} = rand \\ c_1 = \tan[\pi(rand - 1/2)] \end{cases} \quad (43)$$

whereas $rand$ is any number within the range of (0, 1)

Following rule updates every particle in each iteration

$$P'_k = P_{k0} + c_1(P_{k0} - P_{best}^G) \quad (44)$$

where P_{k0} is the calculated position of k particle. After w_{max} iterations, P_{best}^G is the global best position with the ideal fitness and P'_k is the position of k after the update. During the next iteration, the position of the k -th particle is

$$P_k = \begin{cases} P'_k, f(P_{k0}) > f(P'_k) \\ P_{k0}, f(P'_k) > f(P_{k0}) \end{cases} \quad (45)$$

When c_1 is positive, P'_k will be far from P_{best}^G and when c_1 is negative, P'_k will be near P_{best}^G . By using CM operators, half of the particles will disperse away from the center to discover a better position. Equating the updated position with the earlier one, the outcome with better fitness remains. This will not only help get better solutions but enhance the diversity of the swarm as well.

(2) Landmark operators

In the classic PSO, at every iteration during the landmark operator phase, half while also converging to the center of the swarm decrease the swarm population. This will result in the early convergence of the PSO. To prevent this, we replace the old landmark operator with a Cauchy one, which only updates the particle based on its optimal position. The Cauchy function is written as

$$F(x; 0, 1) = \frac{2}{\pi} \arctan(x), x \in (0, +\infty) \quad (46)$$

CM operator c_2 is written as

$$\begin{cases} \frac{2}{\pi} \arctan(c_2) = rand \\ c_2 = \tan\left(\frac{2}{\pi} rand\right) \end{cases} \quad (47)$$

In each iteration, every particle updated by the following rule

$$P_{k0}^{w_{min}} = P_{k0}^{w_{max}} + c_2(P_{best}^G - P_{k0}^{w_{max}}) \quad (48)$$

where $P_{k0}^{w_{max}}$ is the position of k th particle after w_{max} iteration, while P_{best}^G is the global best position.

During the CM landmark operator phase, every particle steadily moves towards the global best outcome because of the operator c_2 . An effective CM operator can ensure the steady convergence of the algorithm.

Hybrid algorithm

This section offers the comprehensive steps of the hybrid algorithm for the multi-UAV swarm formation:

Step 1: Design the 3D mission environment, and terrain model according to equations (1) and (2). Set the origin and ending points for the UAV swarm.

Step 2: Initialize the parameters including total particle number S , particle position and velocity D , inertia weight w , Cauchy weight coefficients c_1 , c_2 , maximum number of iterations T .

Step 3: Design the fitness function using equations (3) to (11). Run the hybrid algorithm once.

Step 4: Observe the fitness functions of the whole swarm and calculate the UAV's best position and formation's best position.

Step 5: Update the constraints inertia weight w using equation (40) and Cauchy weight coefficients c_1 , c_2 using equations (43) and (47).

Step 6: Update the particles using proposed methods including swarm update using equation (39) and Cauchy mutant operators using equations (44) and (48).

Step 7: Replicate steps 4–6 until the hybrid algorithm achieves the maximum iterations T .

Step 8: Finish the hybrid algorithm and plot the fitness functions.

Table 1 presents the pseudo-code for the proposed method, which we can implement in any programming language or MATLAB.

Simulation and discussion

To prove the effectiveness of the designed technique, this section presents the simulations with different environments. The simulations run on a computer with a processor of Intel® Core™ i7-1065G7, Win 10 64-bit operating system, 16 GB RAM. The software used for simulations is MATLAB® R2020a.

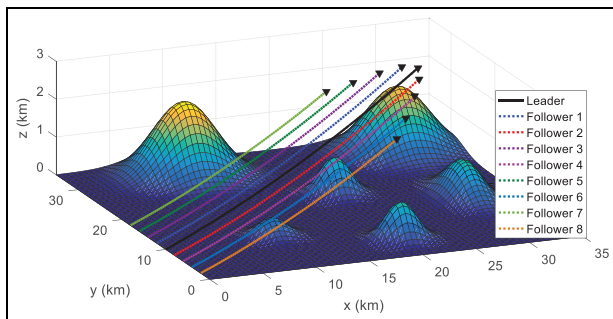
This paper tested the proposed algorithm in two different scenarios. We chose these scenarios to best represent the most common challenges that both the civilian and military forces face during their day-to-day operations. The first scenario

Table 1. Pseudo code.

```

// Mission area
1 Set flying space by Eq. (1);
2 Set terrain by Eq. (2);
3 Set starting and ending positions for 9 UAVs
//Hybrid algorithm initialization
4 Set the swarm parameters  $S, D, w_{max}, w_{min}, c_1, c_2, T$ ;
5 Set  $UAVnumber$  as 9;
6 Initialize parameters of the algorithm;
7 Set a relatively large value for fitness function;
8 Calculate particle fitness value and select  $P_{best}$  and  $P_{best}^G$ 
//Main loop
9 For  $i = 1$  to  $UAVnumber$ 
10 For  $j = 1$  to  $T$ 
11 Update swarm parameters by Eqs. (34), (37), and (41);
12 Update swarm particles by Eqs. (33), (38), and (42);
13 Calculate particle fitness value;
14 Sort fitness value of all particles;
15 Select  $P_{best}$  and  $P_{best}^G$ ;
16 Remember the value and the iteration for  $P_{best}^G$ ;
17  $j = j + 1$ 
18 End
19  $i = i + 1$ 
20 End
//Output
21 Output the fitness function for the swarm formation

```

**Figure 4.** 3D environment.

contains an environment that includes mountainous terrain. The second scenario includes a more complex dynamic environment with radars and obstacles.

Scenario 1

In this simulation scenario, a formation of nine UAVs is to fly from the origin to target under a mountainous terrain. Six

mountains are present in this scenario, whose constraints are given in Table 2.

The swarm has four important objectives here; to avoid the mountains, to avoid other UAVs, to maintain the formation, and to take the shortest possible route from the starting position to the destination.

Figure 4 shows the 3D environment with mountains of different peaks. We can see the path of the swarm from start to finish while avoiding the mountains and other UAVs. While Figure 5 shows the same environment in the xy-plane, Figure 6 presents the xz-plane, and Figure 7 shows the yz-plane.

Figure 8 presents the fitness of all the UAVs through multiple iterations. Since we have to show the fitness levels of nine UAVs, the graph is a bit hard to read, and hence, a zoomed-in portion of the graph is also present. As we can see, the leader takes the longest to converge because it also travels the most distance.

Comparing the fitness graph with a similar technique presented in Zhou et al. (2016), we can see that the fitness levels for the traditional PSO and their proposed method QPSO are in the range of 71–72. While our designed method performs significantly better with fitness levels in the range of 20–30.

Finally, Table 3 presents the starting position, destination, and the distance each UAV traveled during the whole mission. We can notice the different origin points for the UAVs are to avoid a collision.

Scenario 2

In this simulation scenario, a swarm of nine UAVs is to fly from the origin to target in an environment with radars and obstacles. Three obstacles are present in this scenario whose constraints are given in Table 4. While Table 5 presents the placements of the four radars.

In this scenario, the swarm has five important objectives; to avoid the radar detection radius, to avoid the obstacles, to avoid other UAVs, to maintain the formation, and to take the shortest possible route from the starting position to the destination.

Figure 9 shows two different views of the three-dimensional environment with the radars and the obstacles. We can see the path of the formation from start to finish. We can also observe how it avoids radars and obstacles. The formation is hard to maintain in this scenario because of dynamic threats, therefore, we see slight variations in the formation as compared to scenario 1. Figure 10 shows the mission environment in xy-plane, while Figure 11 presents the yz-plane.

Table 2. Constraints of the mountains.

No.	1 st mountain	2 nd mountain	3 rd mountain	4 th mountain	5 th mountain	6 th mountain
Height (km)	0.5	1	1.75	1	1	2
Centre position (x,y)	(10,10)	(20,20)	(30,30)	(20,5)	(30,15)	(10,30)
Decline along axis (x)	2	2	4	2	2	4
Decline along axis (y)	2	2	4	2	4	4

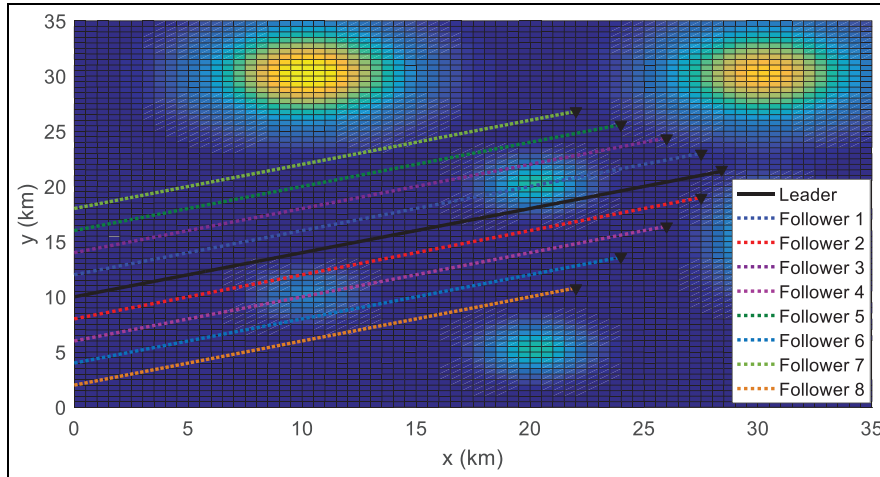


Figure 5. Environment in xy-plane.

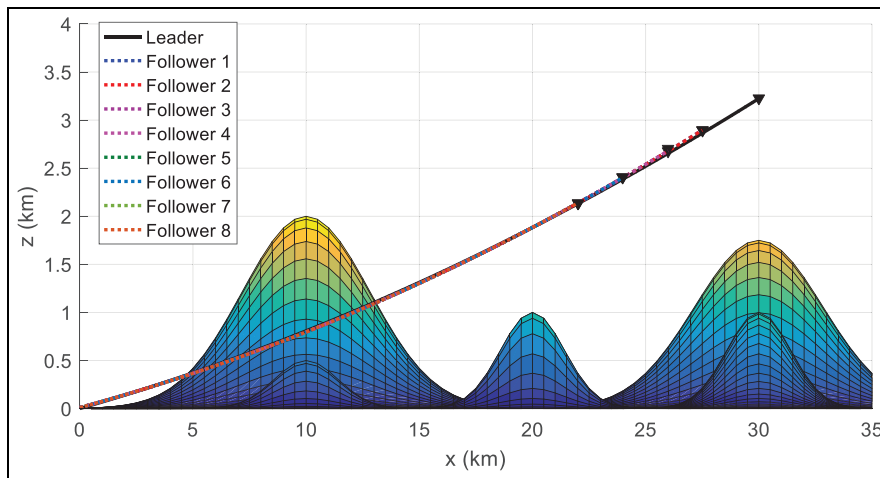


Figure 6. Environment in xz-plane.

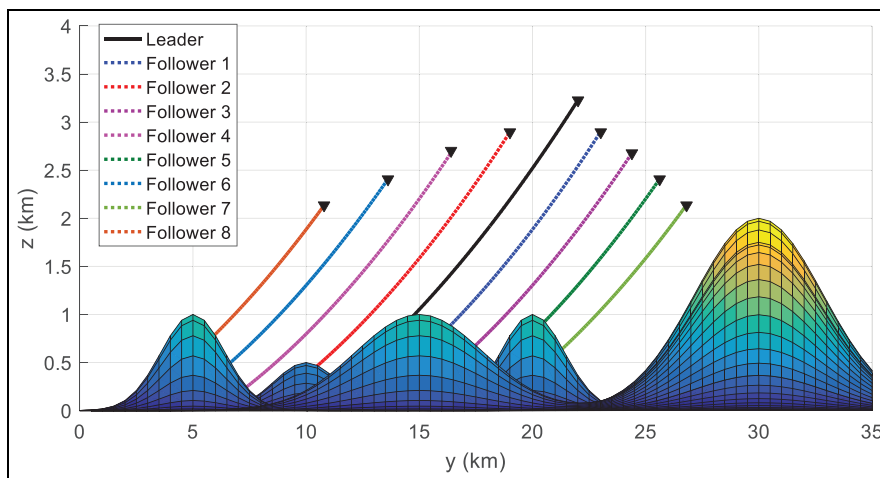


Figure 7. Environment in yz-plane.

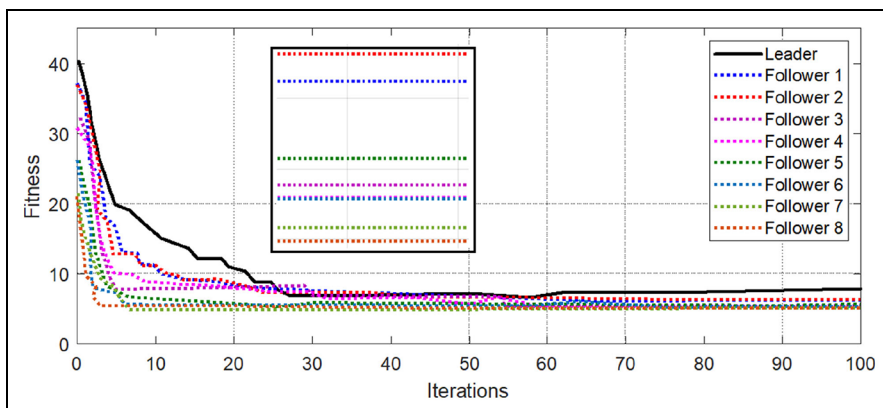


Figure 8. Fitness of all UAVs in scenario I.

Table 3. Coordinates of UAVs in scenario I.

	Leader UAV	Follower UAV 1	Follower UAV 2	Follower UAV 3	Follower UAV 4	Follower UAV 5	Follower UAV 6	Follower UAV 7	Follower UAV 8
Origin (x,y,z)	(0,10,0)	(0,12,0)	(0,8,0)	(0,14,0)	(0,6,0)	(0,16,0)	(0,4,0)	(0,18,0)	(0,2,0)
Target (x,y,z)	(30,22,3.2)	(25,22,2.5)	(25,18,2.5)	(25,24,2.5)	(25,16,2.5)	(25,26,2.5)	(25,14,2.5)	(25,28,2.5)	(25,12,2.5)

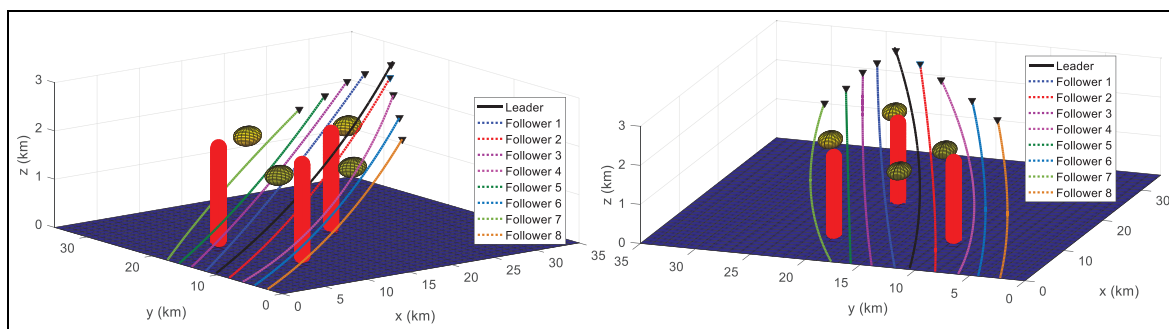


Figure 9. 3D mission area through different angles.

Table 4. Constraints of the obstacles.

	1 st obstacle	2 nd obstacle	3 rd obstacle
Position (x,y)	(8,20)	(10,10)	(20,18.6)
Altitude	2	2	2

Table 5. Radar placements.

	1 st radar	2 nd radar	3 rd radar	4 th radar
Centre	1.5	1.5	1.5	1.5
Radius (x,y,z)	(19,14,1.5)	(23,20,2)	(13,22,3.2)	(11.5,15.5,1.5)

Figure 12 shows the fitness of all the UAVs during multiple iterations. Since we have to show the fitness levels of nine UAVs on a single graph, it is a bit hard to comprehend, and hence, the zoomed-in portion of the graph is also present. Along with the leader, four follower UAVs also take longer

than other UAVs to converge because of the complexity of the mission environment.

Finally, Table 6 presents the starting position, destination, and the distance each UAV traveled during the whole scenario. Again, the origin points for all UAVs are different to avoid a collision.

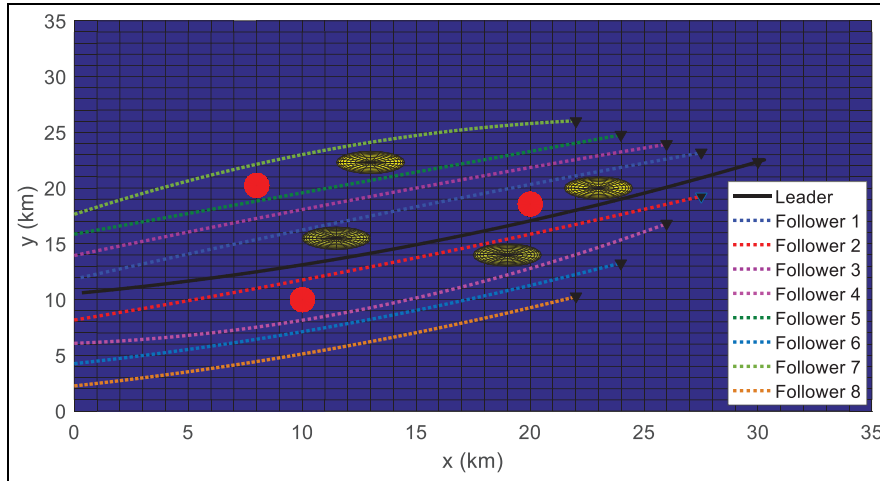


Figure 10. The mission area in xy-plane.

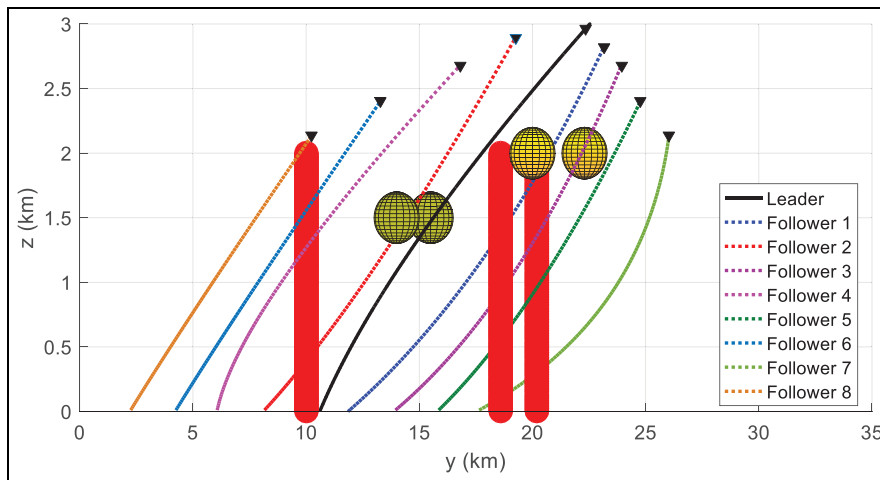


Figure 11. The mission area in yz-plane.

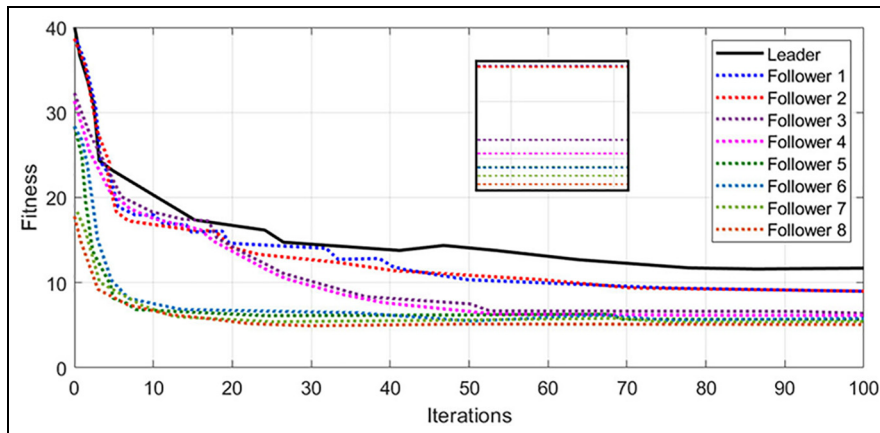


Figure 12. Fitness of all UAVs in scenario 2.

Table 6. Coordinates of UAVs in scenario 2.

	Leader UAV	Follower UAV 1	Follower UAV 2	Follower UAV 3	Follower UAV 4	Follower UAV 5	Follower UAV 6	Follower UAV 7	Follower UAV 8
Origin (x,y,z)	(0,10,0)	(0,12,0)	(0,8,0)	(0,14,0)	(0,6,0)	(0,16,0)	(0,4,0)	(0,18,0)	(0,2,0)
Target (x,y,z)	(33,24.5,3.3)	(29,23.5,3)	(25,18.5,2.5)	(25,23.5,2.5)	(25,16,2.5)	(25,24,2.5)	(25,14,2.5)	(25,26,2.5)	(25,12,2.5)

Conclusion and future research

For the formation control of the multi-UAV swarm in mountainous terrain, the paper first modeled the terrain, radar, and collision cost. Afterward, the study discussed the PSO algorithm and its characteristics. Then, the research introduced Cauchy mutant operators and designed a novel hybrid algorithm. This algorithm was an improvement over classic PSO and offered much better fitness. The simulations demonstrate that the designed hybrid algorithm could find the shortest possible paths with better fitness while also avoiding the obstacles and other UAVs. The algorithm helped maintain the formation and met coordination parameters.

Future research could focus on further enhancing the fitness function and reducing the terrain, radar, and collision costs. The authors also propose to implement the designed algorithm on hardware and perform some experiments. Then, the experimental results and the simulation results presented in this study can be compared to see if there is a variation in performance.


Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research is supported by China National Nature Science Foundation with grant number 61374165.

ORCID iD

Zain Anwar Ali  <https://orcid.org/0000-0002-2143-2879>

References

- Cetin O and Yilmaz G (2016) Real-time autonomous UAV formation flight with collision and obstacle avoidance in unknown environment. *Journal of Intelligent & Robotic Systems* 84(1–4): 415–433.
- Chen Y, Luo G, Mei Y, et al. (2014a) UAV path planning using artificial potential field method updated by optimal control theory. *International Journal of Systems Science* 47(6): 1407–1420.
- Chen Y, Yu J, Su X, et al. (2014b) Path planning for multi-UAV formation. *Journal of Intelligent & Robotic Systems* 77(1): 229–246.
- Cheng R and Jin Y (2015) A social learning particle swarm optimization algorithm for scalable optimization. *Information Sciences* 291: 43–60.
- Das PK, Behera HS and Panigrahi BK (2016) A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning. *Swarm and Evolutionary Computation* 28: 14–28.
- Duan H and Luo Q (2015) New progresses in swarm intelligence-based computation. *International Journal of Bio-Inspired Computation* 7(1): 26–35.
- Duan Z and Shen J (2017) Synchronization problem of 2-D coupled dynamical networks with communication delays and missing measurements. *Multidimensional Systems and Signal Processing*: 30(1) 39–67.
- Duan Z, Zhai G and Xiang Z (2016) Exponential consensus for hierarchical multi-agent systems with switching topology and inter-layer communication delay. *IET Control Theory & Applications*, 10(4): 451–460.
- He L, Bai P, Liang X, et al. (2018) Feedback formation control of UAV swarm with multiple implicit leaders. *Aerospace Science and Technology* 72: 327–334.
- Huang C and Fei J (2018) UAV path planning based on particle swarm optimization with global best path competition. *International Journal of Pattern Recognition and Artificial Intelligence* 32(6): 1859008.
- Huang Y, Wang H and Yao P (2016) Energy-optimal path planning for solar-powered UAV with tracking moving ground target. *Aerospace Science and Technology* 53: 241–251.
- Jabbarpour M, Malakooti H, Noor R, et al. (2014) Ant colony optimisation for vehicle traffic systems: applications and challenges. *International Journal of Bio-Inspired Computation* 6(1): 32–56.
- Kothari M and Postlethwaite I (2013) A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees. *Journal of Intelligent & Robotic Systems* 71(2): 231–253.
- Liu Y, Zhang X, Guan X, et al. (2016) Adaptive sensitivity decision based path planning algorithm for unmanned aerial vehicle with improved particle swarm optimization. *Aerospace Science and Technology* 58: 92–102.
- Marble J and Bekris K (2013) Asymptotically near-optimal planning with probabilistic roadmap spanners. *IEEE Transactions on Robotics* 29(2): 432–444.
- Olfati-Saber R (2006) Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control* 51(3): 401–420.
- Pehlivanoglu Y (2012) A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV. *Aerospace Science and Technology* 16(1): 47–55.
- Rajput U and Kumari M (2017) Mobile robot path planning with modified ant colony optimisation. *International Journal of Bio-Inspired Computation* 9(2): 106–113.
- Roberge V, Tarbouchi M and Labonte G (2018) Fast genetic algorithm path planner for fixed-wing military UAV using GPU. *IEEE Transactions on Aerospace and Electronic Systems* 54(5): 2105–2117.
- Shakhathreh H, Sawalmeh A, Al-Fuqaha A, et al. (2019) Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. *IEEE Access* 7: 48572–48634.
- Stodola P and Mazal J (2016) Applying the ant colony optimisation algorithm to the capacitated multi-depot vehicle routing

- problem. *International Journal of Bio-Inspired Computation* 8(4): 228–233.
- Sun J, Tang J and Lao S (2017) Collision avoidance for cooperative UAVs with optimized artificial potential field algorithm. *IEEE Access* 5: 18382–18390.
- Tian D and Shi Z (2018) MPSO: Modified particle swarm optimization and its applications. *Swarm and Evolutionary Computation* 41: 49–68.
- Wang J and Xin M (2012) Integrated optimal formation control of multiple unmanned aerial vehicles. *IEEE Transactions on Control Systems Technology* 21(5): 1731–1744.
- Wang X, Yadav V and Balakrishnan SN (2007) Cooperative UAV formation flying with obstacle/collision avoidance. *IEEE Transactions on Control Systems Technology* 15(4): 672–679.
- Xia X, Xing Y, Wei B, et al. (2019) A fitness-based multi-role particle swarm optimization. *Swarm and Evolutionary Computation* 44: 349–364.
- Xuan-Mung N and Hong SK (2019) Robust adaptive formation control of quadcopters based on a leader–follower approach. *International Journal of Advanced Robotic Systems* 16(4): 172988141986273.
- Yaghoobi T and Esmaili E (2017) An improved artificial bee colony algorithm for global numerical optimisation. *International Journal of Bio-Inspired Computation* 9(4): 251–258.
- Yang P, Tang K, Lozano JA, et al. (2015) Path planning for single unmanned aerial vehicle by separately evolving waypoints. *IEEE Transactions on Robotics* 31(5): 1130–1146.
- Yao P, Wang H and Su Z (2016) Cooperative path planning with applications to target tracking and obstacle avoidance for multi-UAVs. *Aerospace Science and Technology* 54: 10–22.
- Zhou S, Kang Y, Dai H, et al. (2016) Multi-UAVs formation autonomous control method based on RQPSO-FSM-DMPC. *Mathematical Problems in Engineering* 2016: 1–14.