IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 September 2022, accepted 28 September 2022, date of publication 10 October 2022, date of current version 14 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3213035

RESEARCH ARTICLE

Game Theory-Based Optimal Cooperative Path Planning for Multiple UAVs

LANH VAN NGUYEN[1], (Member, IEEE), MANH DUONG PHUNG[2], AND QUANG PHUC HA[1], (Senior Member, IEEE)

[1]School of Electrical and Data Engineering, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia
[2]Fulbright University Vietnam, Ho Chi Minh City 700000, Vietnam

Corresponding author: Lanh Van Nguyen (vanlanh.nguyen@uts.edu.au)

The work of Lanh Van Nguyen was supported by the Vingroup Science and Technology Scholarship Program, VinUniversity, Vingroup, for overseas study for master's and doctoral degrees.

ABSTRACT This paper presents new cooperative path planning algorithms for multiple unmanned aerial vehicles (UAVs) using Game theory-based particle swarm optimization (GPSO). First, the formation path planning is formulated into the minimization of a cost function that incorporates multiple objectives and constraints for each UAV. A framework based on game theory is then developed to cast the minimization into the problem of finding a Stackelberg-Nash equilibrium. Next, hierarchical particle swarm optimization algorithms are developed to obtain the global optimal solution. Simulation results show that the GPSO algorithm can generate efficient and feasible flight paths for multiple UAVs, outperforming other path planning methods in terms of convergence rate and flexibility. The formation can adjust its geometrical shape to accommodate a working environment. Experimental tests on a group of three UAVs confirm the advantages of the proposed approach for a practical application.

INDEX TERMS Cooperative path planning, UAV, Stackelberg-Nash game, PSO.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) have rapidly emerged with many interesting applications in both military and civilian domains [1], [2]. To carry out complicated tasks, it is required the teaming of multiple UAVs flying in formation or swarm. The cooperative control of a group of UAVs working together for a robotic task can result in such advantages as high efficiency, reliability, and flexibility, compared to the task execution with single UAVs [3]. Therefore, the problem of path planning for UAV formation control has received great interest. Indeed, path planning plays a significant role in completing flight missions for controlling multiple unmanned vehicle systems. Of importance in formation path planning for UAVs is how to avoid threats while completing a task by taking various constraints and cooperative cohesion conditions into consideration [4]. The strategy here is to plan a flight route for individual UAVs from the starting location to the goal while minimizing the total flight cost.

The associate editor coordinating the review of this manuscript and approving it for publication was Gerardo Flores.

For unmanned vehicles, the commonly-used A* algorithm can be incorporated with a prediction technique and route replanning strategy, or combined with numerical optimal control to solve the path planning problem subject to constraints [5]. The Voronoi graph-based techniques have been introduced by simplifying the space representation [6]. As mainly developed for 2D, the classical A* search and Voronoi techniques would need some extension to cope with cooperative tasks in complex 3D dynamic environments.

The artificial potential field (APF) is another technique for path planning of UAVs [7]. In this approach, the operation space is considered as a potential field that is characterized as ''attractive'' surrounding the target and ''repulsive'' in the neighborhood of obstacles. For UAV cooperative control, additional potential fields are also included to result in attraction effects for maintaining the formation configuration, and repulsion effects for inter-UAV collision avoidance. The paths are then generated when the total force acting on each UAV is induced from all the potential fields at each position. This technique can produce smooth and continuous paths.

On the other hand, it encounters the problem of local minima when the total force is equal to zero.

From the hierarchical perspective, path planning for UAVs can be considered as a high level control problem, whereby optimal and predictive control methods can be used for minimization of a cost function [8]. For multiple UAVs, cooperative path planning can be formulated into an optimization problem for single vehicles subject to multiple constraints [9]. To generate the paths, optimal control can be applied to individual UAVs and then extended to the whole group. Although this approach can solve the optimization problem under different constraints for the formation when performing a robotic task, it involves high computational complexity.

Computational evolution algorithms have been applied to multi-UAV cooperative path planning with the capability to find optimal solutions in complex scenarios [10]. This approach, not requiring discretization of the workplace, can generate smooth cooperative paths for the UAVs. However, if cooperative constraints and maneuvering tasks are not properly addressed, it may converge to sub-optimal solutions.

The key problem of UAV formation control is to resolve possible conflicts and interactions among the leader and followers or among the members of the group to maintain a desired shape under various constraints when executing a robotic task. In this context, the game theory, an important branch of mathematics for studying conflicts and interactions among rational decision-makers [11], can offer a powerful tool to determine an optimal strategy. Indeed, as a solid framework for strategic interactions among competing players, the game theory has found various successful applications, e.g., in distributed energy generations [12], cooperative spectrum sensing [13], or recently in highway maintenance [14].

During a game, to maximize the profit, players can choose to take action depending on not only their own strategy but also of others. Therefore, the best strategy is often decided on what a player expects others to do. A game, in general, can be categorized as cooperative or non-cooperative [15]. In cooperative games (CGs), several players share a common goal to win or to achieve a profit better than by playing alone. A major issue with CGs is trade-offs between stability and efficiency of the overall system [16]. In contrast, players in noncooperative games (NCGs), possessing available information of their own intentions, payoff functions, and procedural details of the game, can pursue their own strategies. For NCGs, each player in a Nash game is equally aware of other players' strategies to reach an equilibrium, which is called the Nash equilibrium. Although having information about the decisions of others, each player has to simultaneously make one's own decision in a symmetric competition, such as to find optimal control parameters [17] or to seek strategies for multiple clusters in a distributed way [18]. On the other hand, in Stackelberg games [19], players have to adopt sequential steps depending on the moves of the leading players. After the leaders take their first actions in priority, the followers can adapt their strategies accordingly.

In this paper, the path planning problem is formulated into a game with UAVs as the players. By incorporating the Stackelberg and Nash game into a two-layer framework with a defined Stackelberg-Nash equilibrium, each UAV in the group can develop a self-enforcing controller to establish and maintain a desired geometric shape. The desired formation is kept in the control layer while constraints on each UAV's path are taken into account via a cost function in the planning layer. Here, the cost function is minimized by using a new algorithm named the game theory-based particle swarm optimization (GPSO), developed for UAV cooperative path planning. Notably, our method considers cooperative constraints on the optimal paths of individual UAVs in the group to cover all requirements on formation, path feasibility, flexibility, and safety. For this, a hierarchical particle swarm optimization (PSO) algorithm is developed to determine the strategy for each UAV to reach the Stackelberg-Nash equilibrium. The resulting optimal paths can be achieved while maintaining the formation. Unlike other formation control methods, the generated paths here can be self-adjusted to reconfigure the whole group so that they can better adapt to changes in the operating environment.

The contributions and innovations of our work are threefold: (i) comprehensively formulating the cooperative path planning problem for multiple UAVs into the minimization of a multi-objective cost function, (ii) casting the solution of the minimization problem into the search for the equilibrium of a two-layer Stackelberg-Nash game, and (iii) developing new algorithms using game theory-based particle swarm optimization (GPSO) for finding optimal solutions. The results of this work enable the effective deployment of flexible and cooperative formations of UAVs to accommodate complex scenarios to effectively improve the autonomy of UAV-based task execution under harsh, hazardous, and maybe hostile conditions.

This paper is organized as follows. The cooperative path planning problem is formulated in Section II. The proposed Stackelberg-Nash game and hierarchical PSO algorithms are introduced in Section III. Section IV provides numerical simulation results. Section V presents the setup and experimental validation results, followed by Section VI for the conclusion.

II. PROBLEM FORMULATION

Consider a fleet of UAVs flying in a known environment with obstacle models prescribed by cylinders as illustrated in Figure 1, wherein the inertial frame, xyz, is defined relative to the sea level with the z axis pointing upward. The flight formation has a hierarchical leader-follower structure, as illustrated in Figure 2. The structure is organized into l layers, $l = 1, 2, \ldots, \Lambda$. This structure, having advantages over the behavior-based or virtual ones in maintaining the stability of the formations under different environments, is quite popular in multi-robot coordination and often used for data acquisition and communication among UAVs [20], [21]. We denote V_l a UAV at layer l and consider it as the leader of N_{l+1} followers at layer $l+1$, which are, in turn, denoted as $V_{(l+1)n}$,

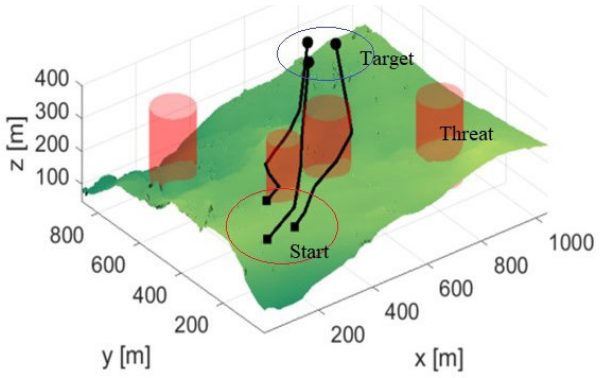


FIGURE 1. Illustration of the UAV path planning problem.

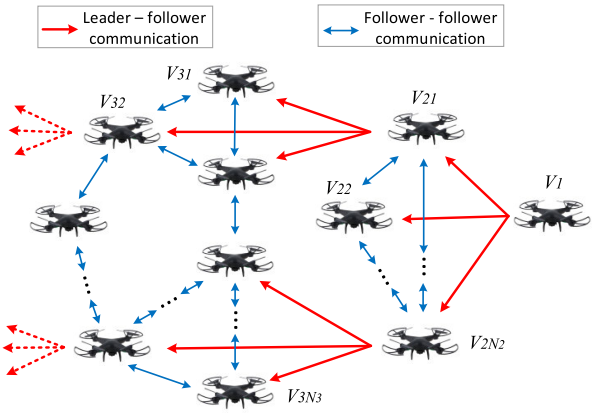


FIGURE 2. Hierarchical leader-follower structure of UAV team.

$n = 1, 2, \ldots, N_{l+1}$. The location of the formation is defined by $P = [{P_1}^T, {P_2}^T, \ldots, {P_S}^T]^T$, where S is the total number of UAVs in the team and $P_n = (x_n, y_n, z_n)^T$ is the position of the n-th vehicle, $n = 1, 2, \ldots, S$. To define the formation shape, a set of reference positions $P_r = [{P_1}_r{}^T, {P_2}_r{}^T, \ldots, {P_S}_r{}^T]^T$ is given, where $P_{n_r} = (x_{n_r}, y_{n_r}, z_{n_r})^T$ is the reference position of the n-th UAV. The desired vector between two neighbors n and n' is computed as $P_{n_r n'_r} = P_{n_r} - P_{n'_r}$.

In a cooperative path planning problem, the aim is to find optimal paths for all UAVs from their starting points to target locations, fulfilling all requirements imposed by constraints on formation shape, path length, threat avoidance, and turning angle limit. We approach this problem with a cost function for each UAV in the team by incorporating those constraints to formulate the cooperative path planning into an optimization problem. The cost function associated with UAV_n thus has the form:

$$J(X_n) = \sum_{i=1}^{\eta} \omega_{ni} J_i(X_n), \tag{1}$$

where X_n represents the path of UAV_n, $J_i(X_n)$ is the cost corresponding to constraint i, ω_{ni} is a weighting factor, and η is the number of constraints. The path X_n is defined by a set of K nodes, represented by waypoints $P_n(k) = (x_n(k), y_n(k), z_n(k))^T, k = 1, 2, \ldots, K$, that connect the flight path of UAV_n. The cost function $J_i(X_n)$, $i = 1, 2 \ldots, \eta$, for each constraint is determined in the following.

A. FORMATION CONSTRAINT

The formation constraints are determined from the desired structure of the geometric shape and interactions among UAVs. Using the graph theory, we define UAV_n as a vertex v_n and its interconnection with $\text{UAV}_{n'}$ as an edge $\epsilon_m = (v_n, v_{n'})$ of a directed graph $\mathcal{G}$. Let $\mathcal{V} = \{v_1, v_2, \ldots, v_S\}$ be the set of vertices and $\mathcal{E} = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_M\}$ be the set of edges, the graph $\mathcal{G}$ can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. To establish the formation, the graph must be connected, i.e., for any two vertices $(v_n, v_{n'}) \in \mathcal{V}$, there exists an interconnection between them, or an edge in $\mathcal{E}$. This interconnection is weighed by $\mu_{nn'}$ as our graph is an edge-weighted graph. The incidence matrix $\mathcal{D}$ of the graph has the dimension of $S \times M$, where its element is equal to 1 if the UAV is the head of an edge, -1 if the UAV is the tail of an edge, and 0 otherwise.

The formation error for edge $(v_n, v_{n'})$ is computed from $P_n - P_{n'} - P_{n_r n'_r}$. The total formation error can be expressed via the incidence matrix as

$$\begin{aligned} E &= \sum_{n,n' \in \mathcal{E}} \mu_{nn'} \|P_n - P_{n'} - P_{n_r n'_r}\|^2 \\ &= (P - P_r)^T \hat{\mathcal{D}} \hat{W} \hat{\mathcal{D}}^T (P - P_r) \\ &= ||P - P_r||^2_{\hat{\mathcal{D}} \hat{W} \hat{\mathcal{D}}^T}, \end{aligned} \tag{2}$$

where $\hat{W} = W \otimes I_3$ and $W = \text{diag}[\mu_{nn'}]$ is a diagonal weight matrix of dimension $M \times M$, $\hat{\mathcal{D}} = \mathcal{D} \otimes I_3$, in which operator $\otimes$ is the Kronecker product.

Define $\mathcal{L} = \mathcal{D} W \mathcal{D}^T$ as the Laplacian of the graph $\mathcal{G}$, which is symmetric and positive semi-definite, $\hat{\mathcal{L}} = \hat{\mathcal{D}} \hat{W} \hat{\mathcal{D}}^T = \mathcal{L} \otimes I_3$ is also symmetric and positive semi-definite [22]. The formation error can be rewritten as $E = ||P - P_r||^2_{\hat{\mathcal{L}}}$. The cost function coming from the formation constraint for UAV_n is then defined as

$$J_1(X_n) = \sum_{k=1}^{K} ||P(k) - P_r||^2_{\hat{\mathcal{L}}_n}. \tag{3}$$

Apart from maintaining the formation, it is also important to avoid intervehicle collisions among the UAVs. Let $\bar{d}_n(k)$ be the Euclidean distance from UAV_n to its nearest neighbor at waypoint k, i.e.,

$$\bar{d}_n(k) = \min_{n' = \{1,2,\ldots,S\}, n' \neq n} \|P_n(k) - P_{n'}(k)\|, \tag{4}$$

r_n and d_s be respectively the radius of UAV_n and its safety clearance. To avoid collisions, the distance between a UAV to its nearest neighbor should then be greater than the sum of the clearance d_s and double of the UAV radius r_n. Accordingly, the cost function (3) is revised as,

$$J_1(X_n) = \sum_{k=1}^{K} E_n(k), \tag{5}$$

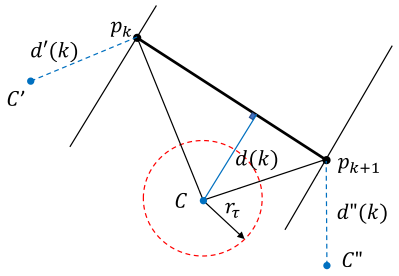


FIGURE 3. Distance between an obstacle and two waypoints.

$$E_n(k) = \begin{cases} ||P(k) - P_r||^2_{\hat{\mathcal{L}}_n}, & \text{if } \bar{d}_n(k) > d_s + 2r_n \\ \infty, & \text{if } \bar{d}_n(k) \le d_s + 2r_n, \end{cases} \quad (6)$$

where the infinity assigned to a cost function implies an infeasible solution so that node k, at risk of collision, should not be considered by an optimizer.

B. PATH LENGTH COST

When planning a path, it is required that its length is minimized to save time and energy, especially for a low-cost UAV. In autonomous operations, a path typically includes a list of waypoints uploaded to the UAV as references for the flight controller to track [23]. With K waypoints, it can be represented by a set of $K-1$ line segments connecting the waypoints as shown in Fig.1. The path length is then simply the sum of those segments. Denoting $P_n(k)$ as waypoint k of path n, the cost representing the length of path n is then computed as:

$$J_2(X_n) = \sum_{k=1}^{K-1} ||P_n(k+1) - P_n(k)||^2. \quad (7)$$

C. OBSTACLE AVOIDANCE

During operation, each UAV needs to avoid collision with not only other UAVs but also obstacles in its working environment, such as trees or buildings. In this work, each obstacle is considered as a threat modeled by a cylinder with radius r_τ.

Let p_k of coordinates (x_k, y_k) be the projection of waypoint $P(k)$ on the Oxy-plane, the distance $d(k)$ from the center of the obstacle $C(x_c, y_c)$ to the path segment k is determined as the shortest distance from C to $p_k p_{k+1}$. If $\|p_{k+1} - p_k\| \neq 0$, $d(k)$ is calculated as the normal distance from point C to the line connecting two points p_k and p_{k+1} as shown in Figure 3, or the smaller distance to those points in the case they are located at one side of the obstacle. Thus, $d(k)$ is computed as

$$d(k) = \begin{cases} \|C - p_k\|, \text{ if } \|p_{k+1} - p_k\| = 0, \\ \dfrac{|(x_{k+1} - x_k)(y_k - y_c) - (x_k - x_c)(y_{k+1} - y_k)|}{\sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}}, \\ \quad \text{if } \|p_{k+1} - p_k\| \neq 0 \text{ and } \mathcal{A} \ge 0, \\ \min(\|C - p_k\|, \|C - p_{k+1}\|), \text{ otherwise}, \end{cases} \quad (8)$$

where

$$\mathcal{A} = [(x_{k+1} - x_k)(x_c - x_k) + (y_{k+1} - y_k)(y_c - y_k)] \times [(x_k - x_{k+1})(x_c - x_{k+1}) + (y_k - y_{k+1})(y_c - y_{k+1})].$$

The threat cost $J_3(X_n)$ is then calculated across waypoints $P_n(k)$ for $\mathcal{T}$ obstacles as below:

$$J_3(X_n) = \sum_{k=1}^{K-1} \sum_{\tau=1}^{\mathcal{T}} D_\tau(k),$$

where $D_\tau(k)$

$$= \begin{cases} 0, & \text{if } d(k) > d_s + r_\tau + r_n \\ (d_s + r_n + r_\tau) - d(k), & \text{if } r_n + r_\tau < d(k) \\ & \le d_s + r_n + r_\tau \\ \infty, & \text{if } d(k) \le r_n + r_\tau. \end{cases} \quad (9)$$

D. ALTITUDE CONSTRAINT

In many applications, such as surface inspection, it is essential that the UAVs fly within a certain altitude range limited by the minimum and maximum heights, h_{min} and h_{max}, respectively. Let $h_n(k)$ be the relative height of the UAV with respect to the ground. The desired flying height for it is then $\bar{h} = 0.5(h_{min} + h_{max})$. Thus, the altitude cost can be computed as

$$J_4(X_n) = \sum_{k=1}^{K} H_n(k),$$

$$\text{where } H_n(k) = \begin{cases} |h_n(k) - \bar{h}|, & \text{if } h_{min} \le h_n(k) \le h_{max}, \\ \infty, & \text{otherwise.} \end{cases} \quad (10)$$

E. SMOOTHNESS

Due to motion restraints of UAV dynamics, they are unable to make a sharp turn, and thus it is required for the algorithm to generate smooth paths. This can be achieved by limiting changes in the turning and climbing angles.

The turning angle between two consecutive segments, $\theta_n(k)$, is computed as

$$\theta_n(k) = cos^{-1}\left(\frac{\overrightarrow{p_n(k)}.\overrightarrow{p_n(k+1)}}{\|\overrightarrow{p_n(k)}\|.\|\overrightarrow{p_n(k+1)}\|}\right), \quad (11)$$

where $\overrightarrow{p_n(k)} = Proj_{/Oxy}\{P_n(k+1) - P_n(k)\}$ is the projection of segment $(P_n(k+1) - P_n(k))$ on the plane Oxy.

The climbing angle, $\varphi_n(k)$ between the path segment $(P_n(k+1) - P_n(k))$ and $\overrightarrow{p_n(k)}$ is calculated as:

$$\varphi_n(k) = tan^{-1}\left(\frac{z_n(k+1) - z_n(k)}{\|\overrightarrow{p_n(k)}\|}\right). \quad (12)$$

The smoothness cost can then be defined as

$$J_5(X_n) = \beta_1 \sum_{k=1}^{K-2} \theta_n(k) + \beta_2 \sum_{k=1}^{K-1} |\varphi_n(k) - \varphi_n(k+1)|, \quad (13)$$

where β_1 and β_2 are the penalty coefficients of the turning and climbing angles, respectively.

III. GPSO DEVELOPMENT

Given the cost function $J(X_n)$ defined for each UAV, the cooperative path planning becomes finding paths X_n,

$n = 1, \ldots, S$, to simultaneously minimize $J(X_n)$. Since the value of $J(X_n)$ depends on the path X_n generated for UAV$_n$ itself as well as other paths of the remaining UAVs in the team, finding optimal solutions remains a challenging problem. The game theory concept of resolving conflicts and handling interactions has been well-recognized [24]. Aligning well with cooperative path planning in robotics, it can be promising for a game-theoretic framework to solve the formation control problem with the development of a suitable optimization tool. To deal with the problem of cooperative path planning for UAVs, we propose here a game theory-based particle swarm optimization (GPSO) approach consisting of two steps. In the first step, a Stackelberg-Nash game is formulated from the cooperative path planning problem. A hierarchical PSO-based algorithm is then developed to solve for the Stackelberg-Nash equilibrium to obtain the optimal paths.

A. STACKELBERG-NASH GAME FOR COOPERATIVE PATH PLANNING

We first introduce some definitions.

Definition 1: Each UAV in the group is defined as a decision-maker or player. Hence, the leading and following UAVs are considered respectively as the leading and following decision-makers.

Definition 2: A strategy of player UAV$_n$ is its path X_n.

Definition 3: The payoff for player UAV$_n$ is its cost function $J(X_n)$.

1) STACKELBERG GAME

The Stackelberg game is a model aiming to resolve the asymmetric competition among a leading decision-maker and some following decision-makers. In this game, the leader conducts his movement first. The followers then decide their strategies to respond to the leader's decision [25]. Here, the Stackelberg game can be used to model interactions among the UAVs.

Now, the Stackelberg game for UAVs can be described as $G_S = \big((L, F), (S_l, S_f), (J_l, J_f)\big)$, where (L, F) is a set of players with leading player L and following players F defined as a subset $F = (F_1, F_2, \ldots, F_N)$. The pair (S_l, S_f) stands for the strategy sets of the leading player, S_l, and followers, S_f. They are defined respectively as $S_l = (X_{l_1}, X_{l_2}, \ldots, X_{l_\Sigma})$, where X_{l_σ} is the decision strategy made by the leader $(\sigma = 1, 2, \ldots, \Sigma)$, and $S_f = (S_{f_1}, S_{f_2}, \ldots, S_{f_N})$, where $S_{f_n} = (X_{n_1}, X_{n_2}, \ldots, X_{n_\Sigma})$ represents all Σ decision strategies made by the n-th follower. The set (J_l, J_f) is the players' payoffs.

Let $\overset{*}{X}_l$ and $\overset{*}{X}_f = (\overset{*}{X}_{f_1}, \overset{*}{X}_{f_2}, \ldots, \overset{*}{X}_{f_N})$ be respectively the best strategy of the leader and of the followers, the Stackelberg equilibrium is defined as $\overset{*}{S}_s = (\overset{*}{X}_l, \overset{*}{X}_f)$, which satisfies correspondingly

$$J_f(X_l, \overset{*}{X}_f(X_l)) \le J_f(X_l, X_f(X_l)), \tag{14a}$$

$$J_l(\overset{*}{X}_l, \overset{*}{X}_f(\overset{*}{X}_l)) \le J_l(X_l, \overset{*}{X}_f(X_l)). \tag{14b}$$

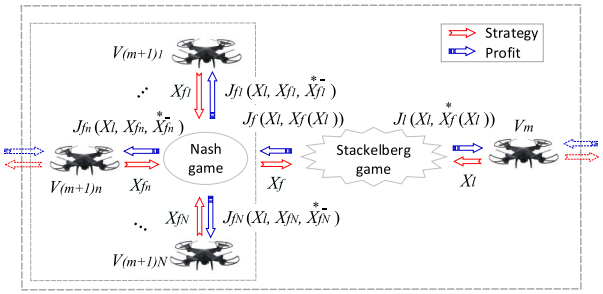


FIGURE 4. Stackelberg-Nash game illustration for UAV formation.

The relation $X_f(X_l)$ represents the strategy X_f of the following players as a function of the leader's strategy X_l. From (14a) and (14b), $\overset{*}{S}_s = (\overset{*}{X}_l, \overset{*}{X}_f)$ can be obtained as

$$\overset{*}{X}_f = \arg\min J_f \quad \text{s.t.} \quad X_f \in S_f, \tag{15a}$$

$$\overset{*}{X}_l = \arg\min J_l \quad \text{s.t.} \quad X_l \in S_l. \tag{15b}$$

2) NASH GAME

Apart from interactions between the leader and followers, those among the followers should also be taken into account. We use the Nash game for their modeling, making use of the symmetry in the roles of the players [26]. In a Nash equilibrium, each player is assumed to know the best strategies of other rivals, and no player can gain more payoff by changing only their own plan. The Nash equilibrium thus provides an approach to obtain optimal results for all symmetric players, which represent the following UAVs. The Nash game can be expressed as $G_N = \big(F, S_f, J_f\big)$. The Nash equilibrium is defined as $\overset{*}{S}_f = (\overset{*}{X}_{f_1}, \overset{*}{X}_{f_2}, \ldots, \overset{*}{X}_{f_N})$, satisfying the following condition:

$$\forall X_{n_\sigma} \in S_{f_n},\ J_{f_n}(\overset{*}{X}_{f_n}, \overset{*}{X}^{-}_{f_n}) \le J_{f_n}(X_{n_\sigma}, \overset{*}{X}^{-}_{f_n}), \quad n = \{1, 2, \ldots, N\}, \sigma = \{1, 2, \ldots, \Sigma_n\}, \tag{16}$$

where $\overset{*}{X}^{-}_{f_n} = (\overset{*}{X}_{f_1}, \ldots, \overset{*}{X}_{f_{n-1}}, \overset{*}{X}_{f_{n+1}}, \ldots, \overset{*}{X}_{f_N})$ is the optimal strategy set of X_n's rivals. The Nash equilibrium is obtained as

$$\overset{*}{X}_{f_n} = \arg\min_{X_{f_n}} J_{f_n}(X_{f_n}, \overset{*}{X}^{-}_{f_n}). \tag{17}$$

3) STACKELBERG-NASH GAME

By combining the two above models, the cooperative path planning problem for multiple UAVs can be represented by a Stackelberg-Nash game as illustrated in Figure 4. The game is expressed as

$$G = \big((L, F), (S_l, S_f), (J_l, J_f), (F, S_f, J_f)\big). \tag{18}$$

The Stackelberg-Nash equilibrium, $\overset{*}{S} = (\overset{*}{X}_l, \overset{*}{X}_{f1}, \overset{*}{X}_{f_2}, \ldots, \overset{*}{X}_{f_N})$, is defined to meet the conditions:

$$J_{f_n}(X_l, \overset{*}{X}_{f_n}(X_l), \overset{*}{X}^{-}_{f_n}(X_l)) \le J_{f_n}(X_l, X_{f_n}(X_l), \overset{*}{X}^{-}_{f_n}(X_l)), \quad \forall X_{n_\sigma} \in S_{f_n},\ n = \{1, 2, \ldots, N\}, \sigma = \{1, 2, \ldots, \Sigma_n\}, \tag{19a}$$

$$J_l(\overset{*}{X}_l, \overset{*}{X}_f(\overset{*}{X}_l)) \le J_l(X_l, \overset{*}{X}_f(X_l)). \tag{19b}$$

To find $\overset{*}{S}$, a hierarchical PSO-based algorithm is developed as described in the following.

B. HIERARCHICAL PSO FOR STACKELBERG-NASH GAME

From the above, the UAV cooperative path planning is reduced to finding the strategy $\overset{*}{S}$ of a Stackelberg-Nash game that fulfills all requirements of conditions (19a) and (19b) to bring the game to its equilibrium. However, simultaneously solving inequalities (19a) and (19b) is challenging and even impractical for analytical methods as involving non-differentiable cost functions $J(X_n)$ and dependent variables $X_f(X_l)$. Instead, heuristic optimization techniques based on swarm intelligence are more viable and computationally efficient for this problem. Among heuristic optimization techniques, PSO has been widely used for path planning problems as being effective in the optimal search. This is owing to the balance between exploration and exploitation of swarm particles [23], [27], [28]. The main advantages of PSO include its robustness, fast convergence, and computational efficiency, making it thus suitable for optimizing interactions among players in a game theory framework. In this paper, a hierarchical optimization algorithm based on PSO is developed to find the best strategies for the Stackelberg-Nash equilibrium. Those strategies represent the optimal paths of the UAVs.

1) PSO AND LEADER-FOLLOWER PATH PLANNING

Particle swarm optimization (PSO) is a population-based method that exploits a population of individuals to probe promising regions of the search space for optimal solutions. In this context, the population is called a swarm, and the individuals are called particles. Each particle moves with an adjustable velocity to find its best position. Besides, the best position of the swarm is shared among all individuals to navigate their direction.

Consider a D-dimensional search space, $\mathcal{S} \in R^D$, and a swarm consisting of N_p particles. The position and velocity of particle i are D-dimensional vectors denoted as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})^T$ and $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})^T$, respectively. The best position encountered by particle i is a point in $\mathcal{S}$ denoted as $Q_i = (q_{i1}, q_{i2}, \ldots, q_{iD})^T$. Let g be the particle that has the best position among all individuals in the swarm at time t. The evolvement of the swarm is updated by the following equations:

$$V_i(t+1) = c_0 V_i(t) + c_1 r_1 [Q_i(t) - X_i(t)] + c_2 r_2 [Qg(t) - X_i(t)], \quad (20)$$

$$X_i(t+1) = X_i(t) + V_i(t+1), \quad (21)$$

where c_0 is an inertia factor, c_1 and c_2 are, respectively, the cognitive and social coefficients, and r_1 and r_2 are random samples uniformly distributed in the interval [0, 1].

For path planning with PSO, we encode the position of a particle by flight path X_n. The whole swarm thus includes N_p candidate path solutions and their evolvement results in the optimal solution. To better explore the search space, we use the recently-developed spherical vector-based particle swarm optimization (SPSO) algorithm [23]. The SPSO uses spherical coordinates to describe the nodes of a flight path so that it can exploit the correspondence between spherical variables and maneuverable parameters of the UAV to speed up the search process. Here, we propose a game theory-based hierarchical approach to extend the SPSO for cooperative path planning involving multiple UAVs during the search for the Stackelberg-Nash equilibrium.

Since there are conflicts and interactions among the leader and followers' strategies within these two objectives expressed in (19a) and (19b), the proposed GPSO algorithm to obtain the overall Stackelberg-Nash equilibrium also covers two loops in general. The pseudo-code for the whole hierarchy of leader-follower optimization is described in Algorithm 1. Therein, to incorporate the interactions within the hierarchy, from line 1 to line 6 of the inner loop, optimal strategies for the followers are first obtained by solving the Nash equilibrium, wherein the leader's strategy X_l remains unchanged. The optimal strategies of the followers $\overset{*}{X}_f(X_l)$ obtained are then used to find the best strategy for the leader at the outer loop overall from line 1 to line 9. Further explanations of both the inner and outer loop are presented respectively in Algorithms 2 and 3 in the following.

Algorithm 1 Leader-Follower Optimization Hierarchy

1. Fix X_l;
for $n = 1 : N$ **do**
  for $n = 1 : N$ **do**
    2. Fix $X_{f_n}^-$;
    3. $\overset{*}{X}_{f_n}(X_l, X_{f_n}^-) = \arg\min_{X_{f_n}} J_{f_n}(X_l, X_{fn}, X_{f_n}^-)$;
    4. Obtain $\overset{*}{X}_{f_n}(X_l, X_{f_n}^-)$ and $\overset{*}{X}_{f_n}^-(X_l, X_{f_n})$;
  end for
  5. Substitute $X_{f_n}^- = \overset{*}{X}_{f_n}^-(X_l, X_{f_n})$ into $\overset{*}{X}_{f_n}(X_l, X_{f_n}^-)$;
  6. Obtain $\overset{*}{X}_{f_n}(X_l, \overset{*}{X}_{f_n}^-)$;
end for
7. Obtain $\overset{*}{X}_f(X_l)$;
8. $\overset{*}{X}_l = \arg\min_{X_l} J_l(X_l, \overset{*}{X}_f(X_l))$;
9. Obtain $\overset{*}{X}_l$;
10. Substituting $X_l = \overset{*}{X}_l$ into $\overset{*}{X}_f(X_l)$;
11. Obtain $\overset{*}{X}_f(\overset{*}{X}_l)$;

2) HIERARCHICAL GAME-THEORETIC PSO IMPLEMENTATION

As per the optimization hierarchy designed for the Stackelberg-Nash game equilibrium, PSO algorithms are developed to implement the framework and obtain optimal solutions. The implementation procedure consists of four steps: Nash game initialization, inner-loop optimization, Stackelberg game initialization, and outer-loop optimization.

a: NASH GAME INITIALIZATION

Initially, the parameters of the PSO and random strategies of the followers are generated at a given strategy X_l of the leader. The followers' payoffs are then optimized to obtain their best

strategies, $\overset{*}{X}_f(X_l)$, with respect to the leader strategy. At this stage, the PSO algorithm is implemented in the inner loop, corresponding to the Nash game for followers. In this inner loop, parameters of the PSO such as weights c_{0f}, c_{1f}, c_{2f}, number of iterations $maxIt_f$, and number of particles $nPop_f$ for the followers are first selected for initialization.

b: INNER LOOP OPTIMIZATION FOR FOLLOWER STRATEGIES

After having the leader's strategy X_l and initializing random strategies for the followers, the cost of follower n at each iteration can be described as $J_{f_n}(X_l, X_{f_n}(It_f), \overset{*}{X}^{-}_{f_n}(It_f - 1))$, where $\overset{*}{X}^{-}_{f_n}(It_f - 1)$ is the best strategies of the other followers previously. By minimizing this cost, the optimal strategy $\overset{*}{X}_{f_n}(It_f)$ of player n can be achieved correspondingly. Based on $\overset{*}{X}_{f_n}(It_f)$ recorded after each iteration, the strategy of the remaining followers is then adjusted according to swarm dynamics (20) and (21). Eventually, the inner loop optimization process will yield the best strategies $\overset{*}{X}_f(X_l)$ of all followers for strategy X_l of the leader. At the termination of the inner loop, the Nash equilibrium is obtained for all the followers since none of the players can gain benefit just by altering its own strategy. The pseudo-code for the optimization implementation to achieve the Nash equilibrium is presented in Algorithm 2.

The followers' optimal strategies $\overset{*}{X}_f(X_l)$ resulting from the Nash game are then fed to the outer loop for Stackelberg game optimization.

Algorithm 2 Inner-Loop PSO Implementation for Followers Path

Require: Leader's strategy X_l;

1. Initialize PSO parameters: $c_{0f}, c_{1f}, c_{2f}, maxIt_f, nPop_f$;
2. Set $It_f = 0$, generate random follower's strategies;
3. Obtain the initial optimal follower's strategies $\overset{*}{X}_f(It_f)$;

for $It_f = 1 : maxIt_f$ **do**

4. Recall $\overset{*}{X}_f(It_f - 1)$;
5. Calculate $J_{f_n}(X_l, X_{f_n}(It_f), \overset{*}{X}^{-}_{f_n}(It_f - 1))$, for $n = 1, 2, \ldots, N$;
6. Record $\overset{*}{X}_{f_n}(It_f)$;
7. Update $X_{f_n}(It_f)$;

end for

8. Obtain $\overset{*}{X}_f(X_l)$;

c: STACKELBERG GAME INITIALIZATION

The GPSO algorithm is initiated with a search map and path planning information depending on the task requirements and operating environment. The swarm for the outer loop of the optimization process are initialized with parameters including weights c_{0l}, c_{1l}, c_{2l}, number of iterations $maxIt_l$, and population $nPop_l$.

d: OUTER LOOP OPTIMIZATION FOR LEADER STRATEGY

From the optimal strategies of the followers $\overset{*}{X}_f(X_l)$, the payoff $J_l(X_l, \overset{*}{X}_f(X_l))$ is obtained to evaluate the leader's profit. It is then used to adjust the leader strategy according to equations (20) and (21) of the PSO. This strategy is then fed back to Algorithm 2 to start a new cycle in the inner loop for optimization. The optimization process terminates when the maximum number of iterations is exceeded or no profits are gained. Finally, the best strategies $\overset{*}{X}_f(\overset{*}{X}_l)$ of the followers with respect to the best leader strategies are obtained as an overall result of the game theory-based optimization.

The pseudo-code for the optimization process to achieve the Stackelberg-Nash equilibrium is presented in Algorithm 3.

Algorithm 3 Outer-Loop PSO Implementation for Leader Path

Require: Search map and initial path planning information;

1. Initialize PSO parameters: $c_{0l}, c_{1l}, c_{2l}, maxIt_l, nPop_l$;
2. Set $It_l = 0$, generate random leader's strategies, X_l;

for $It_l = 0 : maxIt_l$ **do**

3. Run **Algorithm 2** to obtain $\overset{*}{X}_f(X_l)$;
4. Calculate the the leader's profit, $J_l(X_l, \overset{*}{X}_f(X_l))$;
5. Record $\overset{*}{X}_l(It_l)$;
6. Update $X_l(It_l)$;

end for

7. Obtain $(\overset{*}{X}_l, \overset{*}{X}_f)$.

The converged outcomes eventually present the global Nash-Stackelberg equilibrium because the leader's strategy is optimized in the outer loop, which covers optimal strategies of all followers in the inner loop given a path of the leader. However, some local optima may be obtained if the results neither converge to a single point nor have oscillatory behavior. In that case, the Nash-Stackelberg equilibrium does not exist. Since the leader's and followers' strategies are generated randomly at the first iteration, Algorithm 1 should be run several times to obtain histograms for the converged cost values, for example, subject to a given threshold.

C. SPEED PROFILES FOR UAV COOPERATION

From Algorithms 1-3, safe and optimal paths can be generated for all vehicles of the formation, defined by a set of K nodes as discussed above. Since distances between two arbitrary nodes and the total path lengths of the UAVs could be different, they would not reach waypoints and targets at the same time by using a constant speed to establish the formation. Therefore, after proceeding with the proposed GPSO hierarchy to obtain waypoints, it is required to apply suitable speed profiles for the UAVs to reach the waypoint and target positions. To achieve this goal, velocity is used as an independent variable along with the generated paths. The following algorithm is therefore developed to compute the required speed profile for each UAV path.

Let $T_k, k = 1, 2, \ldots, K+1$, be the time required for a UAV to travel over the flight segment k. For timely cooperation and synchronization, T_k should be the same for all UAVs. This

leads to the calculation of speed profiles as presented in the pseudo-code of Algorithm 4. Given a reference speed, v_{ref}, the reference time T_k required to complete the longest path segment, l_{max_k}, among all the k-th segments is first determined. The flight speed, $v_n(k)$, of UAV$_n$ for path segment k is then calculated proportionally to the segment length, i.e. $v_n(k) = l_n(k)/T_k$, where $l_n(k)$ is the length of path segment k for UAV$_n$.

Algorithm 4 Speed Profiles Calculation

Require: Path of all vehicles X_n, for $n = 1, 2, \ldots, S$

1. Initialize the reference speed v_{ref};
2. Calculate path segment lengths for all paths: $l_n(k)$;

for $k = 1 : (K + 1)$ **do**

3. Find $l_{max_k} = \max\{l_1(i), l_2(i), \ldots, l_S(i)\}$;
4. Calculate $T_k = l_{max_k}/v_{ref}$;
5. Calculate speed profiles for all UAVs: $v_n(k) = l_n(k)/T_k$;

end for

6. Return speed profiles for all UAVs: $v_n(k)$.

IV. SIMULATION RESULTS

This section presents the simulation results of the proposed algorithm. Due to similarities in the multi-layer hierarchical structure of UAVs, we conduct simulations for a general layer to evaluate the path planning performance. Generally, a layer consists of one leader V_m and several followers $V_{(m+1)n}$, $n = 1, 2, \ldots, N$. In the simulation, we consider two commonly used formation shapes, including the equilateral triangle and diamond formations [29], [30]. However, it should be noted that there are no restrictions on the choice of the formation shape.

A. EVALUATION WITH EQUILATERAL TRIANGLE FORMATION

In this scenario, the objective is to generate paths for three UAVs flying in an equilateral triangle formation. The incidence matrix $\mathcal{D}_1$ is thus defined as

$$\mathcal{D}_1 = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \end{bmatrix}. \tag{22}$$

The weights for interconnections are set as $W_m = [1\ 1\ 0]$, $W_{(m+1)1} = [1\ 0\ 0.01]$, and $W_{(m+1)2} = [0\ 1\ 0.01]$. Here, the weights for interconnections between the leader and followers are set higher than the ones between the followers since it is more desirable to maintain distances with the leader than with the followers. The operation space is chosen as 100m×100m×35m in dimension. Obstacles are modeled as cylinders located at (60, 20), (40, 50), (20, 80), and (80, 70) with a radius of 4m. The starting locations of the leader and followers are chosen at $\left[P_m(0),\ P_{(m+1)1}(0),\ P_{(m+1)2}(0)\right] = [15\ 10\ 20;\ 18.66\ 10\ 10;\ 20\ 20\ 20]$. The target locations were located at $\left[P_m(end), P_{(m+1)1}(end), P_{(m+1)2}(end)\right] = [85\ 80\ 90;\ 88.66\ 80\ 80;\ 20\ 20\ 20]$. The formation reference was given according to the target position, i.e. $P_{m_r} = P_m(end)$, $P_{(m+1)1_r} = P_{(m+1)1}(end)$, and $P_{(m+1)2_r} = P_{(m+1)2}(end)$.

The parameters of the hierarchical PSO were set by the trial-and-error as $c_0 = 0.98$ and $c_1 = c_2 = 1.5$. Both the outer and inner loops of the PSO run with 150 particles and 100 iterations. In the total cost function of the leader, weight factors were chosen as $[\omega_{l1},\ \omega_{l2},\ \omega_{l3},\ \omega_{l4},\ \omega_{l5}] = [0.01,\ 10,\ 100,\ 1,\ 1]$ in order to obtain a short distance avoidance-free path. Meanwhile, weight factors in the total cost function of the followers were chosen as $[\omega_{f1},\ \omega_{f2},\ \omega_{f3},\ \omega_{f4},\ \omega_{f5}] = [10,\ 0.01,\ 100,\ 1,\ 1]$ in order to obtain a formation-maintaining avoidance-free path. The number of waypoints for each path was set as $K = 10$, excluding the start and target positions.

To show the merit of the proposed framework using the Stackelberg-Nash game for UAV cooperative path planning, we have compared it with a distributed algorithm based on the Stag Hunt game approach [31], [32]. The paths of the three UAVs are shown in Figure 5. As can be seen in Figure 5a, the triangular formation with the Stackelberg-Nash game is maintained throughout the flight while both inter-vehicle collisions and obstacle collisions can be avoided. The Stackelberg-Nash equilibrium is reached after 50 iterations for the leader and 55 iterations for the followers at the convergence of cost values as depicted in Figure 6a. Compared to the Stag Hunt game, our algorithm converged faster and with smoother paths. The stable convergence can be confirmed in Figure 7 showing histograms of the final cost values in 35 trials.

B. EVALUATION WITH DIAMOND FORMATION

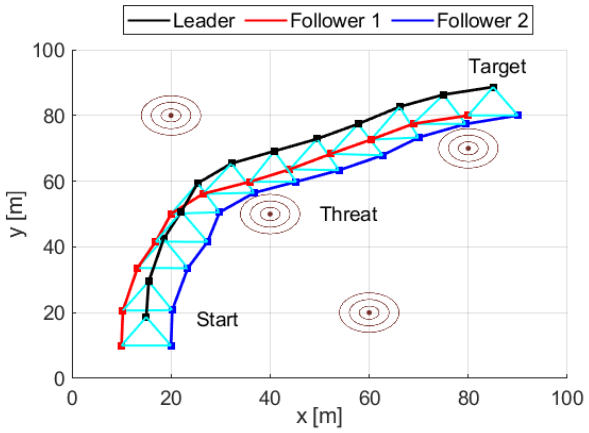
To further evaluate the performance of the proposed cooperative path planning algorithm, we compared it with state-of-the-art methods that considered the group of UAVs as a rigid body, and path planning was obtained for a virtual vehicle located at the centroid of the group [33], [34]. In this comparison, we considered a task in which four UAVs were required to move in a diamond formation. The incidence matrix $\mathcal{D}_2$ was determined as

$$\mathcal{D}_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{bmatrix}. \tag{23}$$

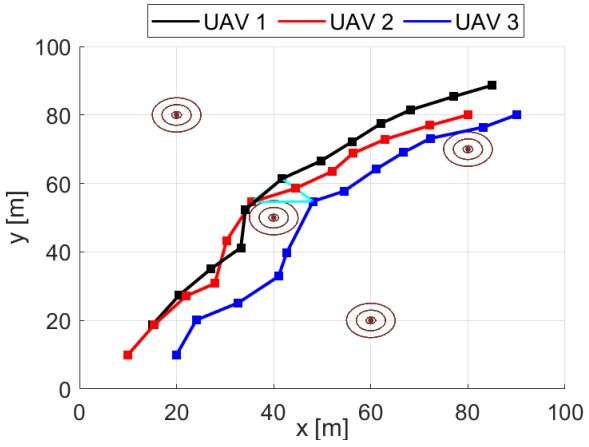
Similarly to the triangular formation case, the weights for interconnections among UAVs were chosen as

$$\begin{aligned} W_m &= \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}, \\ W_{(m+1)1} &= \begin{bmatrix} 1 & 0 & 0 & 0.01 & 0.01 & 0 \end{bmatrix}, \\ W_{(m+1)2} &= \begin{bmatrix} 0 & 1 & 0 & 0.01 & 0 & 0.01 \end{bmatrix}, \\ W_{(m+1)3} &= \begin{bmatrix} 0 & 0.01 & 1 & 0 & 0.01 & 0.01 \end{bmatrix}. \end{aligned}$$

The map used was an area of Christmas Island in Australia obtained from a real digital elevation model (DEM) map [35], as depicted in Figure 8. It has the size of

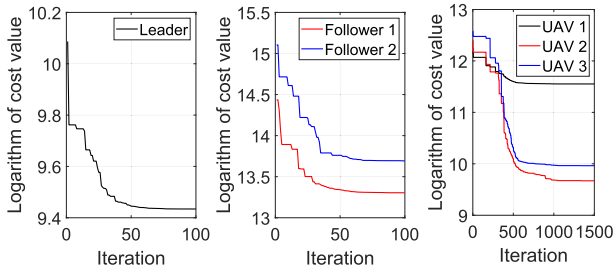


(a) Stackelberg-Nash game



(b) Stag-hunt game

FIGURE 5. Generated formation paths.



(a) Stackelberg-Nash game (b) Stag-hunt game

FIGURE 6. Convergence of cost values.

1000m×1000m×35m. The starting locations were at

$$V_m(0) = [100\ 150\ 20],$$
$$V_{(m+1)1}(0) = [60\ 80.72\ 20],$$
$$V_{(m+1)2}(0) = [140\ 80.72\ 20],$$
$$V_{(m+1)3}(0) = [100\ 11.44\ 20].$$

The target and reference locations were at

$$V_{m_r} = V_m(end) = [850\ 900\ 20],$$
$$V_{(m+1)1_r} = V_{(m+1)1}(end) = [810\ 830.72\ 20],$$

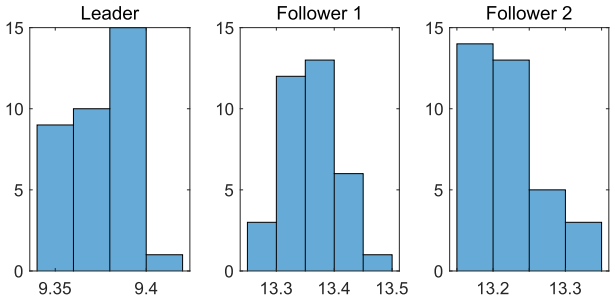
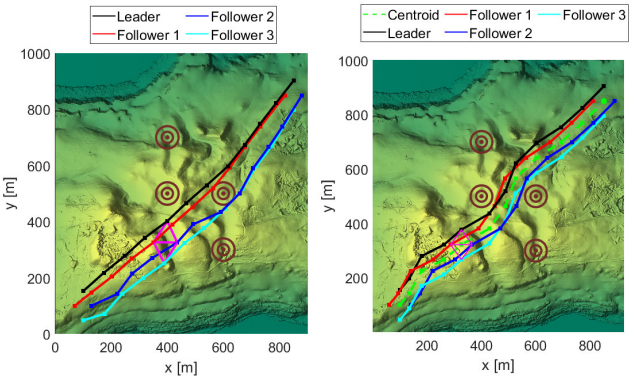


FIGURE 7. Histograms of the final cost values over 35 trials.



(a) Game theory-based formation (b) Rigid body formation

FIGURE 8. Scenario 1.

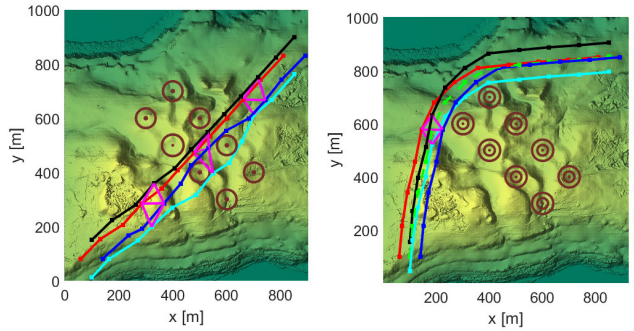
$$V_{(m+1)2_r} = V_{(m+1)2}(end) = [890\ 830.72\ 20],$$
$$V_{(m+1)3_r} = V_{(m+1)3}(end) = [850\ 761.44\ 20].$$

The GPSO parameters are remained as presented in the previous simulation for an equilateral triangle formation.

Comparative results for Scenario 1 with four obstacles can be obtained as shown in Figure 8. It can be seen that both methods, the proposed GPSO and rigid body planning, are able to generate collision-free paths with the formation being well-maintained. However, the game theory-based algorithm is capable of shrinking and enlarging the formation shape depending on the presence of obstacles and thus results in a shorter path, 4391.0 m, compared to the rigid body method, 4397.8 m. This advantage is more prevalent in a more complex situation, Scenario 2, as shown in Figure 9. In this scenario, the UAV formation generated by the rigid body method cannot flight through narrow space between the obstacles so it has to travel a long distance, 5228.0 m, to avoid them (see Figure 9b). The game theory-based method, on the other hand, can change the formation size to generate an optimal path with the length of 4389.1 m.

To evaluate the capability of split and merge of UAV$_n$ at waypoint k, a formation flexibility index (FFI), $\xi_n(k)$, is introduced. It is computed as a ratio of the formation displacement and its corresponding formation reference. For UAV$_n$, it is defined as

$$\xi(k) = \frac{||P(k) - P_r||^2_{\hat{\mathcal{L}}_n}}{||P_r||^2_{\hat{\mathcal{L}}_n}}. \tag{24}$$



(a) Game theory-based formation (b) Rigid body formation

FIGURE 9. Scenario 2.

For the rigid body formation technique, the FFI is zero, i.e., the team of UAVs is inflexible. Meanwhile, a small FFI of less than 0.1 can be obtained for the UAV diamond in Scenario 1. In Scenario 2, the second follower relaxes its formation constraint between the 5-th and 8-th waypoints to avoid obstacles, resulting in the higher FFI of 0.28. This value presents the capability to split and merge the UAV team and further demonstrates the advantage of the proposed approach.

In terms of computational complexity, the proposed method requires running the inner-loop PSO for each leader's strategy to achieve the Stackelberg-Nash equilibrium, resulting in a higher computational cost, particularly for more players and/or with an increased number of constraints. However, since the path planning algorithm is carried out offline, this cost is worth for better overall optimal results.

V. EXPERIMENTAL VALIDATION

In this section, we describe the testbed and the experiments conducted to verify the feasibility and effectiveness of the proposed GPSO algorithm.

A. EXPERIMENTAL SETUP

For experiments, our setup includes three 3DR Solo drones with remote controllers, a ground control station, and communication hardware, as shown in Figure 10. Each drone is equipped with one ARM Cortex A9 processor for running the Arducopter flight operating system and two Cortex M4 168 MHz processors for low-level control. For data acquisition, cameras, laser scanners, or environmental sensors can be attached to the onboard computer of the UAV depending on applications. Communication between the ground control station and the drone is carried out via a private network called 3DR Link Secure WiFi created by modules integrated into the remote controller of the drone [34]. The ground control station used the software named QGround Control to upload the planned path to the drones, fly them autonomously, and download logged data for analysis. In the experiments, the default PID controller was implemented in the 3DR Solo.

The UAV equilateral triangle formation was tested on a park set up with four obstacles as shown in Figure 11. The

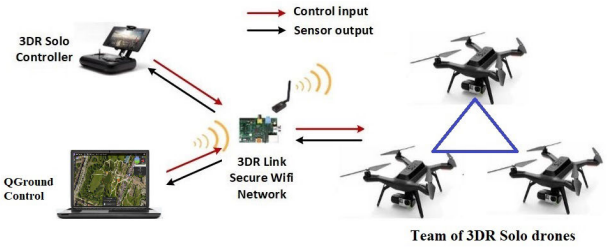


FIGURE 10. Experimental setup.

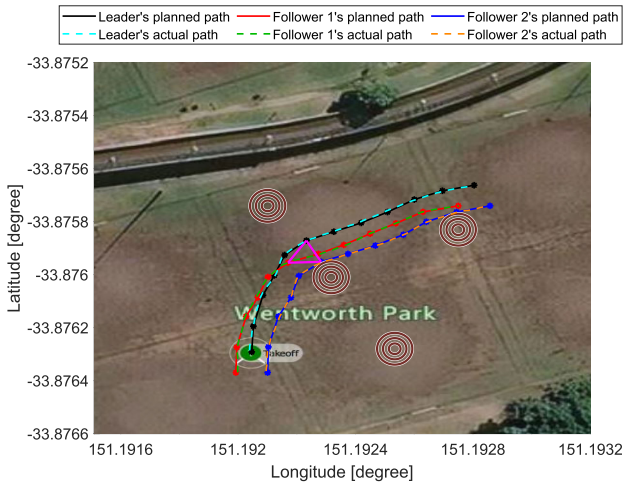


FIGURE 11. Experimental results.

map origin is set at $P_0 = \{-33.87645951; 151.1918842; 0\}$. The relative starting locations of the drones with respect to the origin were set at $[P_m(0), P_{(m+1)1}(0), P_{(m+1)2}(0)] = [15\ 10\ 20; 18.66\ 10\ 10; 20\ 20\ 20]$. The corresponding target and reference positions were chosen as $P_{m_r} = P_m(end) = (85; 88.66; 20)$, $P_{(m+1)1_r} = P_{(m+1)1}(end) = (80; 80; 20)$, and $P_{(m+1)2_r} = P_{(m+1)1}(end) = (90; 80; 20)$. The coordinates of waypoints are first converted into their longitude, latitude, and altitude. The speed profiles of the UAVs are then computed according to Algorithm 4, with $V_{ref} = 1(m/s)$. After that, the waypoints and corresponding speeds are uploaded to the drones using the QGround Control software for their autonomous execution.

B. VERIFICATION RESULTS

Figure 12 shows the drones forming a triangular shape when performing a bridge inspection task. The objectives of path planning and formation maintenance of the three drones are achieved with our proposed GPSO, as shown in the experimental results of Figure 11, in which the solid lines are the planned paths and the dotted lines are the actual paths of the UAV triangle.

To evaluate tracking performance, Figure 13 presents the position errors and speed profiles of the UAVs. It can be seen that all the three UAVs can track their planned path with small tracking errors of 0.4m to 0.6m maximally. In fact, those tracking errors are mainly caused by the GPS positioning

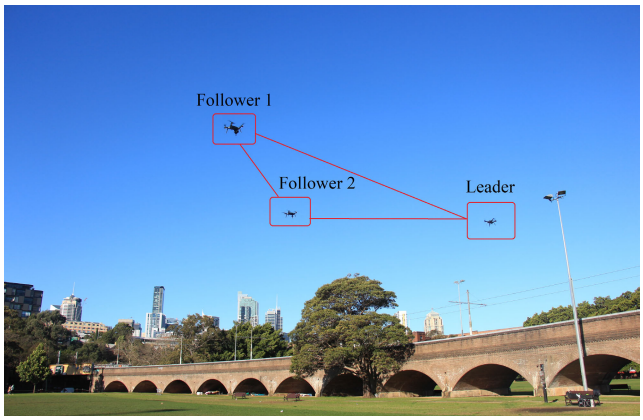


FIGURE 12. UAV triangular formation during experiments.

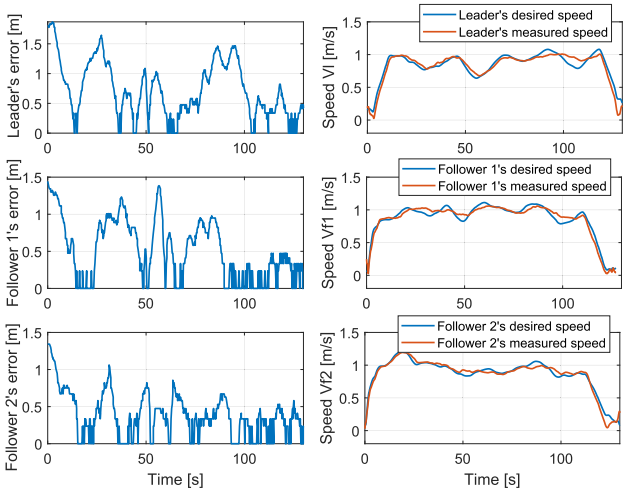


FIGURE 13. Tracking performance.

errors rather than by the tracking controller of the drones themselves. The figure also depicts that all measured speeds accurately track their corresponding desired values generated from the GPSO path planning calculations. This, again, confirms that all UAVs reach the waypoints while maintaining well the formation configuration.

VI. CONCLUSION

In this paper, we have presented a novel method based on the game theory and particle swarm optimization algorithms for the cooperative path planning problem of multiple UAVs navigating in a desired geometric configuration. The UAV collaborative path planning problem is solved by finding the equilibrium of a Stackelberg-Nash game. A hierarchical optimization framework using PSO was integrated to find the game equilibrium by minimizing a global cost function while simultaneously taking into account constraints for the desired shape as well as path length, collision avoidance, altitude, and smoothness. The proposed GPSO cooperative path planning approach can generate a safe path for each UAV in the team with a flexible formation size. Extensive simulation and comparison results are provided for evaluating its performance. Field experiments have also been conducted to demonstrate the feasibility and effectiveness of the proposed method in various scenarios. Our future work will focus on improving the algorithm for online cooperative path planning of UAVs in the presence of dynamic obstacles.

ACKNOWLEDGMENT

Lanh Van Nguyen would like to acknowledge the support by the Vingroup Science and Technology Scholarship Program for Overseas Study for Master's and Doctoral Degrees.

REFERENCES

[1] S. U. Jan and H. U. Khan, "Identity and aggregate signature-based authentication protocol for IoD deployment military drone," *IEEE Access*, vol. 9, pp. 130247–130263, 2021.
[2] S. Rezwan and W. Choi, "Artificial intelligence approaches for UAV navigation: Recent advances and future challenges," *IEEE Access*, vol. 10, pp. 26320–26339, 2022.
[3] E. T. Alotaibi, S. S. Alqefari, and A. Koubaa, "LSAR: Multi-UAV collaboration for search and rescue missions," *IEEE Access*, vol. 7, pp. 55817–55832, 2019.
[4] Y. Liu and R. Bucknall, "Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment," *Ocean Eng.*, vol. 97, pp. 126–144, Mar. 2015.
[5] W. Sheng, B. Li, and X. Zhong, "Autonomous parking trajectory planning with tiny passages: A combination of multistage hybrid A-star algorithm and numerical optimal control," *IEEE Access*, vol. 9, pp. 102801–102810, 2021.
[6] Y. V. Pehlivanoglu, "A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV," *Aerosp. Sci. Technol.*, vol. 16, no. 1, pp. 47–55, 2012.
[7] Z. Pan, C. Zhang, Y. Xia, H. Xiong, and X. Shao, "An improved artificial potential field method for path planning and formation control of the multi-UAV systems," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 1129–1133, Mar. 2022.
[8] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *J. Field Robot.*, vol. 29, no. 2, pp. 315–378, 2012.
[9] A. Bemporad and C. Rocchi, "Decentralized linear time-varying model predictive control of a formation of unmanned aerial vehicles," in *Proc. IEEE Conf. Decis. Control Eur. Control Conf.*, Dec. 2011, pp. 7488–7493.
[10] H. Qu, K. Xing, and T. Alexander, "An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots," *Neurocomputing*, vol. 120, pp. 509–517, Nov. 2013.
[11] R. B. Myerson, *Game Theory: Analysis of Conflict*. Cambridge, MA, USA: Harvard Press, 1997.
[12] M. O. Okoye and H.-M. Kim, "Adopting the game theory approach in the blockchain-driven pricing optimization of standalone distributed energy generations," *IEEE Access*, vol. 10, pp. 47154–47168, 2022.
[13] C. Ji and Q. Qi, "Cooperative spectrum sensing algorithm based on evolutionary game theory," *IEEE Access*, vol. 10, pp. 71557–71565, 2022.
[14] A. Ji, X. Xue, Q. P. Ha, X. Luo, and M. Zhang, "Game theory–based bilevel model for multiplayer pavement maintenance management," *Autom. Construct.*, vol. 129, Sep. 2021, Art. no. 103763.
[15] J. Moura and D. Hutchison, "Game theory for multi-access edge computing: Survey, use cases, and future trends," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 260–288, Aug. 2018.
[16] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
[17] C. Li, H. Zhao, S. Zhen, and Y.-H. Chen, "Control design with optimization for fuzzy steering-by-wire system based on Nash game theory," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 7694–7703, Aug. 2022.
[18] X. Zeng, J. Chen, S. Liang, and Y. Hong, "Generalized Nash equilibrium seeking strategy for distributed nonsmooth multi-cluster game," *Automatica*, vol. 103, pp. 20–26, May 2019.
[19] Y. Li, D. Shi, and T. Chen, "False data injection attacks on networked control systems: A Stackelberg game analysis," *IEEE Trans. Autom. Control*, vol. 63, no. 10, pp. 3503–3509, Oct. 2018.
[20] F. Afghah, A. Razi, J. Chakareski, and J. Ashdown, "Wildfire monitoring in remote areas using autonomous unmanned aerial vehicles," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 835–840.

[21] Z. Hou and I. Fantoni, "Interactive leader–follower consensus of multiple quadrotors based on composite nonlinear feedback control," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 5, pp. 1732–1743, Sep. 2018.
[22] D. Gu, "A differential game approach to formation control," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 1, pp. 85–93, Jan. 2008.
[23] M. D. Phung and Q. P. Ha, "Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization," *Appl. Soft Comput.*, vol. 107, Aug. 2021, Art. no. 107376.
[24] H. Jeong, H. Seo, and H. Kim, "Game theory–based analysis of decision making for coastal adaptation under multilateral participation," *J. Manag. Eng.*, vol. 34, no. 6, Nov. 2018, Art. no. 04018034.
[25] L. Shang and A. M. A. Aziz, "Stackelberg game theory-based optimization model for design of payment mechanism in performance-based PPPs," *J. Construct. Eng. Manag.*, vol. 146, no. 4, Apr. 2020, Art. no. 04020029.
[26] F. Hou, Y. Zhai, and X. You, "An equilibrium in group decision and its association with the Nash equilibrium in game theory," *Comput. Ind. Eng.*, vol. 139, Jan. 2020, Art. no. 106138.
[27] K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 211–224, Jun. 2004.
[28] C. Zhao and D. Guo, "Particle swarm optimization algorithm with self-organizing mapping for Nash equilibrium strategy in application of multiobjective optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 5179–5193, Nov. 2021.
[29] S. Campbell, W. Naeem, and G. W. Irwin, "A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres," *Annu. Rev. Control*, vol. 36, no. 2, pp. 267–283, Dec. 2012.
[30] K. Guo, X. Li, and L. Xie, "Ultra-wideband and odometry-based cooperative relative localization with application to multi-UAV formation control," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2590–2603, Jun. 2020.
[31] J. Du, "An evolutionary game coordinated control approach to division of labor in multi-agent systems," *IEEE Access*, vol. 7, pp. 124295–124308, 2019.
[32] L. V. Nguyen, I. T. Herrera, T. H. Le, D. M. Phung, R. P. Aguilera, and Q. P. Ha, "Stag hunt game-based approach for cooperative UAVs," in *Proc. Int. Symp. Autom. Robot. Construct. (IAARC)*, Jul. 2022, pp. 7488–7493.
[33] Y. B. Chen, J. Q. Yu, X. L. Su, and G. C. Luo, "Path planning for multi-UAV formation," *J. Intell. Robot. Syst.*, vol. 77, no. 1, pp. 229–246, 2015.
[34] M. D. Phung, V. T. Hoang, T. H. Dinh, and Q. P. Ha, "System architecture for real-time surface inspection using multiple UAVs," *IEEE Syst. J.*, vol. 14, no. 2, pp. 2925–2936, Jun. 2020.
[35] *Digital Elevation Model (DEM) of Australia Derived From LiDAR 5 Metre Grid*, Geosci. Australia, Canberra, ACT, Australia, doi: 10.26186/89644.

LANH VAN NGUYEN (Member, IEEE) received the bachelor's degree in electrical engineering from the Thai Nguyen University of Technology (TNUT), Vietnam, in 2011, and the master's degree in control system engineering from the HAN University of Applied Science, the Netherlands, in 2018. He is currently pursuing the Ph.D. degree with the University of Technology Sydney, Australia. He has been a Lecturer and a Researcher with the Faculty of International Training, TNUT, since 2012. His research interests include automation, robotics, and control and navigation for unmanned vehicles.

MANH DUONG PHUNG received the B.Sc. and Ph.D. degrees from Vietnam National University, Hanoi, Vietnam, in 2005 and 2015, respectively. He is currently a Senior Lecturer with Fulbright University Vietnam. His research interests include unmanned aerial vehicles, mobile robot localization and mapping, and swarm intelligence and optimization. He was the organizing committee member of many international conferences and a reviewer of reputable journals of IEEE, Elsevier, and Springer. He was a recipient of several research awards, including the Endeavour Research Fellowship of the Australian Government and the Best paper Award of Australasian Conference on Robotics and Automation.

QUANG PHUC HA (Senior Member, IEEE) received the B.E. degree in electrical engineering from the Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 1983, the Ph.D. degree in complex systems and control from the Moscow Power Engineering Institute, Moscow, Russia, in 1993, and the Ph.D. degree in intelligent systems from the University of Tasmania, Australia, in 1997. He is currently an Associate Professor with the Faculty of Engineering and Information Technology, School of Electrical and Data Engineering, University of Technology Sydney, Australia. His research interests include automation, robotics, and control systems. He has been on the Board of Directors of the International Association of Automation and Robotics in Construction, since 2007. He was a recipient of a number of Best Paper Awards from the IEEE, IAARC, and Engineers Australia, including the Sir George Julius Medal, in 2015. He was the conference chair/the co-chair of several international conferences on automation and intelligent systems. He was on the Editorial Board of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, from 2009 to 2013, *Mathematical Problems in Engineering*, and *Electronics*. He is currently an Associate Editor of *Robotica* (Cambridge Press) and *Automation in Construction* (Elsevier).