

Article

# Collective Motion and Self-Organization of a Swarm of UAVs: A Cluster-Based Architecture

Zain Anwar Ali <sup>1,\*</sup>, Zhangang Han <sup>1,\*</sup> and Rana Javed Masood <sup>2</sup><sup>1</sup> School of Systems Science, Beijing Normal University, Zhuhai 519085, China<sup>2</sup> Department of Electrical Engineering, Usman Institute of Technology, Karachi 75300, Pakistan; rjmasood@uit.edu

\* Correspondence: zainanwar86@hotmail.com (Z.A.A.); zhan@bnu.edu.cn (Z.H.)

**Abstract:** This study proposes a collective motion and self-organization control of a swarm of 10 UAVs, which are divided into two clusters of five agents each. A cluster is a group of UAVs in a dedicated area and multiple clusters make a swarm. This paper designs the 3D model of the whole environment by applying graph theory. To address the aforesaid issues, this paper designs a hybrid meta-heuristic algorithm by merging the particle swarm optimization (PSO) with the multi-agent system (MAS). First, PSO only provides the best agents of a cluster. Afterward, MAS helps to assign the best agent as the leader of the  $n$ th cluster. Moreover, the leader can find the optimal path for each cluster. Initially, each cluster contains agents at random positions. Later, the clusters form a formation by implementing PSO with the MAS model. This helps in coordinating the agents inside the  $n$ th cluster. However, when two clusters combine and make a swarm in a dynamic environment, MAS alone is not able to fill the communication gap of  $n$  clusters. This study does it by applying the Vicsek-based MAS connectivity and synchronization model along with dynamic leader selection ability. Moreover, this research uses a B-spline curve based on simple waypoint defined graph theory to create the flying formations of each cluster and the swarm. Lastly, this article compares the designed algorithm with the NSGA-II model to show that the proposed model has better convergence and durability, both in the individual clusters and inside the greater swarm.

**Keywords:** particle swarm optimization; multi-agent system; vicsek model

**Citation:** Ali, Z.A.; Han, Z.; Masood, R.J. Collective Motion and Self-Organization of a Swarm of UAVs: A Cluster-Based Architecture. *Sensors* **2021**, *21*, 3820. <https://doi.org/10.3390/s21113820>

Academic Editors: Yongwun Jung, Honghai Zhang and Hwangnam Kim

Received: 9 April 2021  
Accepted: 24 May 2021  
Published: 31 May 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In the last decade, research on unmanned aerial vehicles (UAV) has enormously increased due to its awareness with the help of modern social media. On one hand, at the initial level, researchers are taking interest in modeling and controlling a single aerial vehicle. On the other hand, more and more academics are now studying multi-UAV-based scenarios i.e., trajectory tracking, formation control, and path planning of swarms [1–3]. Scientists are studying the natural behavior of a flock of birds, how the birds do the tasks and successfully cooperate within a flock. Researchers in [4–6] took the initiative to design artificial intelligence-based bio-inspired algorithms, i.e., ant colony optimization (ACO), particle swarm optimization (PSO), and pigeon-inspired optimization (PIO) to control robots in formation or swarm.

**Motivation:** The main motivation behind this research is to use the insights gained from the natural behavior of birds and ants and apply them to robots. There are multiple studies available that apply these kinds of bio-inspired algorithms to different kinds of aerial and ground robots [7–9]. Combining a bio-inspired algorithm with a multi-agent system is also a popular idea among researchers nowadays. This study intends to contribute to this growing research area and apply the aforementioned hybrid algorithm to two clusters of UAVs that turn into one big swarm.

**Related Work:** There are multiple research studies concerning the collective motion of a swarm of UAVs and using different approaches to analyze them. In [10], the researchers

used a group of UAVs for fighting forest fires. The researchers in [10] used PSO to help the UAVs plot the most optimal paths toward the fire. In [11], academics applied the PSO for the path planning of a team of UAVs used for 3D surveillance. Teng, H. et al. [11] also presented a fitness function with multiple objectives, including power consumption, flight risk, and area priority. Another study [12] used PSO to plan paths for the formation of multiple UAVs. In [13], researchers used PSO for the trajectory planning of a team of autonomous UAVs. However, classic PSO can get stuck in the local optimum and is slow to converge. This study proposes to use MAS in conjunction with PSO to help reduce the aforementioned issues.

MAS consists of agents that are autonomous and these agents update their position relative to their neighbors' position. MAS has wide-ranging applications from micro-grid control [14] to tracking control of quad-rotor [15]. In [16], a multi-agent system is used to control the formation of multiple UAVs. In [17], researchers used MAS to transport different loads using a group of aircrafts. One study [18] also discussed applications of MAS to fight the COVID-19 pandemic. Mostly, an external entity drives the agents to the target. MAS designates some of the agents as leaders and the rest of the agents start to follow them. Using this method, big swarms can be controlled using few leaders with the help of MAS.

Recently, some studies have explored the hybrid strategy of applying PSO with MAS. In [19], researchers used the combinations of PSO and MAS to optimize the photovoltaic (PV) systems and battery energy storage systems. In [20], researchers treated PSO as a kind of multi-agent system and used this approach for allocating tasks to subgroups within the bigger structure. Biswas, S. et al. [21] presented a hybrid PSO and MAS approach for the path planning and obstacle avoidance of both dynamic and static threats. Similarly, academics proposed a method to defend against distributed denial-of-service (DDoS) attacks using MAS and the agents used the PSO to communicate [22,23].

In [24], Vicsek et al. presented a multi-agent model now known as the Vicsek model. It consists of  $N$  autonomous agents, which have absolute velocities in discrete time. These agents follow the direction of their immediate neighbors. Vicsek et al. [24] concluded that all agents will move in the same direction given their density is large and the external noise is small. This phenomenon is known as synchronization. Recently, scientists have used the Vicsek model for a variety of applications. In [25], researchers used the adaptive Vicsek model to reduce the external disturbances and noise, and retain the formation in a hazardous environment.

Multiple studies use bio-inspired algorithms in multi-agent systems. Leitão, P. et al. [26] used a bio-inspired multi-agent system for enhancing the efficiency of manufacturing systems and industrial automation. Another research [27] showed how a nature-inspired multi-agent system can be effectively used to model pervasive computing systems. In [28], researchers designed a decentralized autonomous robot navigation controller for multi-agent systems. This bio-inspired navigation controller can be used to control a set of robots and allow them to reach the desired location without colliding with static or moving obstacles. In another study [29], researchers presented a bio-inspired multi-agent system for precision agriculture and joint survey missions.

*Contributions:* This research work deals with the self-organization, collective motion, control of a swarm of 10 UAVs, and their definite formation. The following are the major contributions of this research;

- Proposing a hybrid meta-heuristic algorithm by merging the particle swarm optimization (PSO) with the multi-agent system (MAS) along with the dynamic leader selection.
- Designing an algorithm that ensures fast convergence, synchronization, and connectivity between the agents of both clusters.
- Resolving the issues of self-synchronization and collective motion of a swarm of 10 UAVs.
- Validating the performance of the proposed algorithm by using real-time-based numerical simulation, which demonstrates the effectiveness of hierarchical architecture.

*Organization:* The structure of the study is as follows: Section 2 defines the problem formulation and statement of issues as well as the implementation of the hierarchal-based solution framework. Section 3 demonstrates the preliminaries and a complete system model of architecture. It contains the basic concept of terrain environment, allocation of clusters, and the system model. Moreover, Section 4 has the overall design approach of the proposed strategy including the PSO, Vicsek-based MAS model, its synchronization & connectivity, and dynamic leader selection. Section 5 presents the flowchart and the algorithm of the proposed method. Section 6 offers computer-based simulations and discusses their results. Finally, Section 7 concludes the overall article.

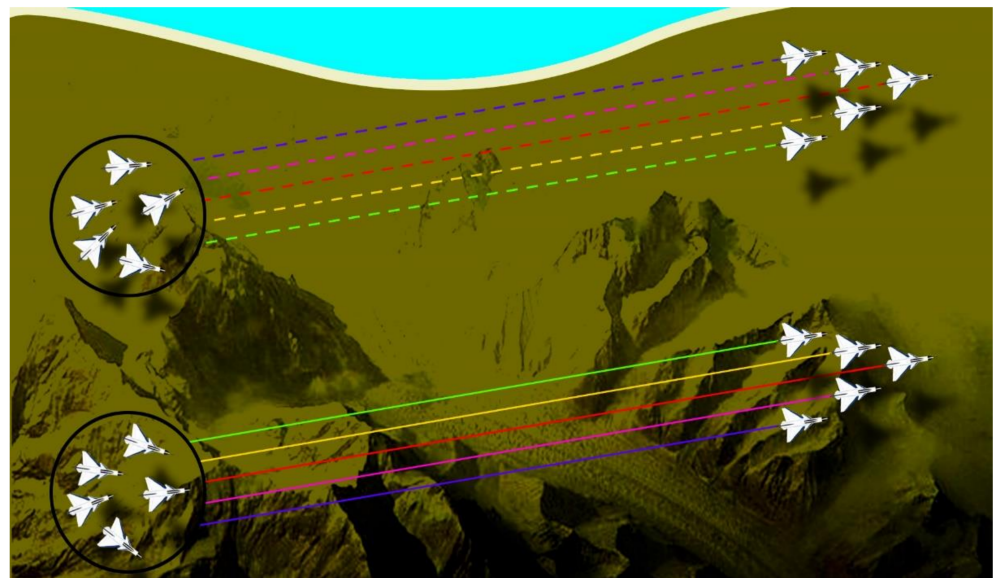
## 2. Problem Description and Solution Architecture

This section presents the issue of the collective motion and self-organization model of a swarm of 10 UAVs. In addition, this section also presents our cluster-based hierarchical solution architecture to solve these issues.

### 2.1. Problem Statements

To accomplish a complicated scenario or task, a group of fixed-wing UAVs sometimes break up into several distinct clusters. Each of the clusters is responsible to perform subtasks. Afterward, when the individual clusters fulfill the desired sub-task, they will merge into a common cluster and start patrolling using the collective motion and self-organizing behavior according to the requirement of the mission. Hence, the formation of the swarms of UAVs can vary over time during the flight. The entire mission is further divided into two scenarios.

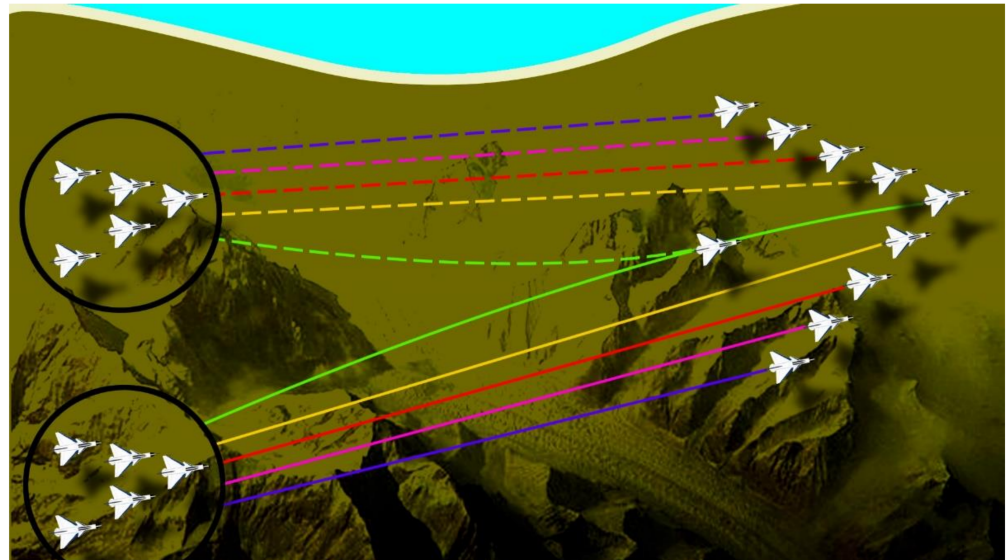
*Problem Statement I:* Figure 1 presents the first scenario. Ten fixed-wing UAVs are divided into two clusters and each cluster has a leader and four followers. It means five UAVs in each cluster collaboratively can perform the subtask. The terrain model contains different obstacles like mountains. Aerial vehicles have a challenging time avoiding the hazardous peaks, maintain their strategy, and reach at the designated area safely. Hence, it is necessary to maintain the desired synchronization and connectivity with their neighbor agents. It is also essential to maintain the leader-follower formation when the UAVs pass through narrow gaps.



**Figure 1.** Scenario 1 containing 10 UAVs divided into two clusters in random positions on the left; the two clusters are in leader-follower formation on the right.

*Problem Statement II:* Figure 2 presents the second scenario. Scenario 2 starts after successfully maintaining the formation of both the clusters and fulfilling their separate

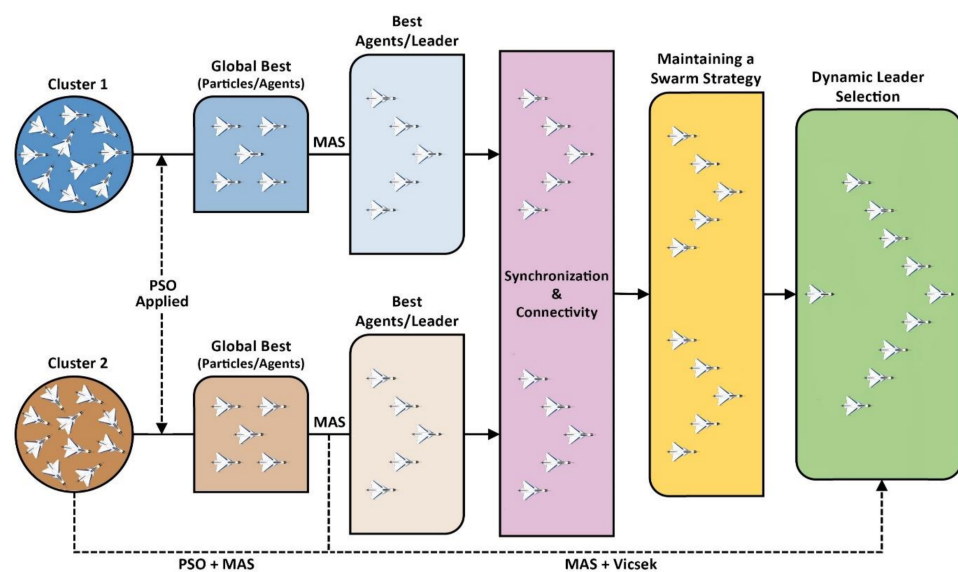
mission tasks in scenario 1. Afterward, both the clusters join in a common cluster called a swarm. At that instant of time, perfect coordination or communication between both the clusters is required to choose which leader will lead the formation of a swarm now. To help achieve the above objective, the algorithm will dynamically select the leader.



**Figure 2.** Scenario 2 containing 10 UAVs divided into two clusters in leader-follower formation on the left, the two clusters merge into one swarm on the right.

## 2.2. Solution Architecture

Figure 3 illustrates the solution architecture of this study. It is evident from the figure that, at first, UAVs were not organized and were in random positions in the cluster. After applying the PSO algorithm in the first phase, the UAVs structure themselves into a strict formation. PSO also determines the best particle of each cluster. In the next step, MAS designates the best agent as the leader of that cluster and the rest of the agents start to follow the leader. Now, in the second phase, Vicsek MAS ensures the synchronization and connectivity between two clusters and also between individual agents within the clusters. Afterward, the two clusters merge to form a swarm and the algorithm dynamically selects the leader of the swarm according to mission requirements.



**Figure 3.** The architecture of the proposed solution.

### 3. Preliminaries and System Model

This section defines the terrain environment and the allocation or division of clusters with the help of graph theory. Next, this section will define the system model, which is followed by the swarm 3D model.

#### 3.1. Terrain Environment

In this study, the 3D landscape used to develop the whole scenario can be given as [12];

$$z(x, y) = \sin(y + d) + e \times \sin(x) + f \times \cos\left(g\sqrt{x^2 + y^2}\right) + h \times \cos(x) + i \times \sin\left(j\sqrt{x^2 + y^2}\right) + k \times \cos(y) \quad (a)$$

In the above equation,  $(d, e, f, g, h, i, j, \text{ and } k)$  are all constants and their values are determined according to the requirement of the terrain. For example, these constants determine the number of peaks in a simulated environment, their heights, and the space between these peaks. The  $(x, y)$  is the horizontal position and  $z$  is the height of its vertical plane.

#### 3.2. Allocation of Cluster

This subsection defines the classical graph theory terminologies and their notations and how they are used in the allocation of clusters [30,31]. An undirected graph is defined by  $G = \langle V, E \rangle$ . This graph contains  $V$ , which is the vertex set, and  $E$ , which is the edge set. These sets are used to define the whole hierarchical architecture and have all the information about the start or takeoff points of UAVs in  $U_T$ .  $U_T$  is a set of UAVs in a cluster and can be defined as  $U_T = (1, 2, \dots, N)$ . Whereas the terrain cost is defined by  $T_c$ . Suppose that  $V = (0, 1, \dots, N)$  is the subset of  $E = (a, b) | a, b \in V, a \neq b$ , where  $(a, b)$  are the neighbors and can also be denoted as  $a \sim b$ . The number of neighbors of each vertex is equal to the number of edges or degree of that vertex. A path of length " $r$ " from vertex  $a$  to  $b$  is a series of  $r + 1$  vertices that start from  $a$  and ends at  $b$  such that the consecutive vertices are adjacent.

For the cluster allocation, the algorithm has to find the main or the final cluster using an optimal route  $N$ th division; where  $N$  is equal to the UAVs clusters, which is decided by  $F_c = (b = 1, 2, \dots, N)$ . Where " $F$ " is the weighing scale of nodes at each cluster that will reach the final cluster " $F_c$ " if it satisfies  $\bigcup_{P=1}^N F_c = F$ . For more details, the reader is referred to [32]. Built by following [32], the issue of dividing the clustering model is rewritten as:

$$\text{Min} \sum_{a=1}^{T_c} \sum_{b=1}^N x_{a,b} d(t_c, c_P) \quad (1)$$

$x_{a,b}$  is the variable for binary decisions.  $x_{a,b}$  is equal to unity if and only if the destination  $t_c$  belongs to the cluster, whose center is  $c_P$ ; otherwise, it will be zero.  $c_P = (b = 1, 2, \dots, N)$  is the center of the cluster and  $d(t_c, c_P)$  is the distance between  $t_c$  and  $c_P$ .

#### 3.3. System Model

Assume  $n$  agents in a cluster, all agents must be able to share their information according to the following [30,31]:

$$n_a = (b | a \sim b) \subseteq (1, \dots, n) \setminus (a) \quad (2)$$

The set of agents which are neighbor to agent  $a$  is  $n_a$ , and the members of the set can strongly link with each other. Now, suppose that there is a predefined radius  $\mathbb{R}$ , which determines the relationship between the neighbors. Now, the position of the agent  $a$ ,  $(a = 1, \dots, n)$  in the universal coordinate system is rewritten as  $(x_a, y_a)$ , and  $V_a = (\dot{x}_a, \dot{y}_a)^T$  is its velocity. The orientation and heading of agent  $a$  is  $\theta_a$  and can be written as:

$$\theta_a = \tan^{-1}(\dot{y}_a, \dot{x}_a) \quad (3)$$

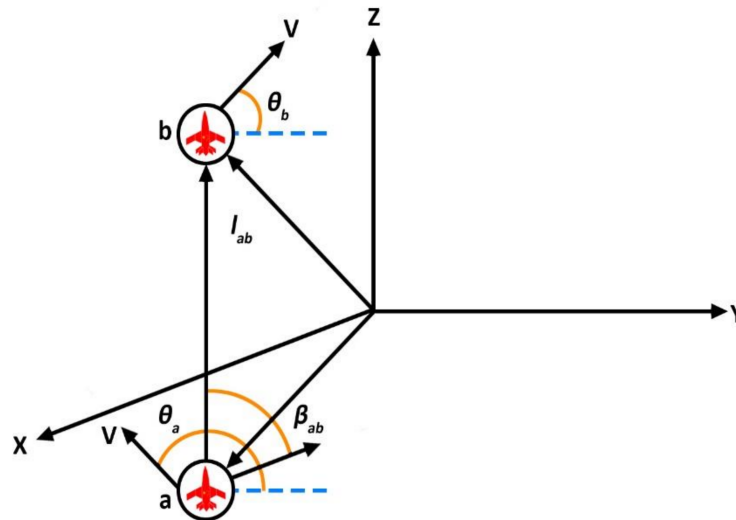
Now it is assumed that all the agents inside the cluster are moving with the constant velocity that is  $V$ . Moreover, suppose a model of the kinematic agent, which is given as:

$$\begin{cases} \dot{x}_a = V \times \cos(\theta_a) \\ \dot{y}_a = V \times \sin(\theta_a) \\ \dot{\theta}_a = \sigma_a \quad a = 1, 2, \dots, n \end{cases} \quad (4)$$

Now the main focus is to control the input  $\sigma_a$ , after that, the cluster of autonomous agents is considered to flock when all the agents manage to have the same velocity, and distances between the agents are steady.

Figure 4 represents the configuration between two agents in a 3D environment inspired by [30]. To adjust the configuration so that it is biologically probable, limit what each agent can do.  $\beta_{ab}$  is the bearing associated with the relative angle among agent  $a$  and agent  $b$  calculated in the coordinates of agent  $a$ . To properly demonstrate the concept of sharing, suppose that individual agent  $a$  can be calculated in terms of ( $\beta_{ab}$ ); the relative bearing in agent-related frame; the change of its bearing rate; and ( $\tau_{ab}$ ) its time to avoid collision concerning any agent  $b$  in the group of neighbors  $n_a$ . Notice that computation of instant collision does not correspond to the calculated relative distance amongst the agents. Now, plain analysis shows that the bearing and relative length between agents  $a$  and  $b$  can be written as:

$$\begin{cases} \beta_{ab} = \tan^{-1}(y_b - y_a; x_b - x_a) - \theta_a + (\pi/2) \\ l_{ab}^2 = (x_b - x_a)^2 + (y_b - y_a)^2 \end{cases} \quad (5)$$



**Figure 4.** The configuration between two agents in a 3D environment revealing the relative distance and bearing between agent  $a$  and  $b$ .

Now, comparable suppositions can be made for the interaction of agents inside a cluster to extend the model presented above to a 3D space. It is also supposed that all agents are moving with constant velocity [30]. The velocity in term of agent  $a$  rewritten as:

$$V_a = [\cos(\theta_a) \sin(\varphi_a); \sin(\theta_a) \sin(\varphi_a); \cos(\varphi_a)]^T \quad (6)$$

In the above equation,  $\theta_a$  and  $\varphi_a$  are the heading and attitude angles, which are written in terms of agent  $a$ . Now the dynamic equation of the agent is;

$$\dot{V}_a = U_{a\theta} X_{a\theta} + U_{a\varphi} X_{a\varphi} \quad (7)$$

The orthonormal vectors are  $X_{a\theta} = (-\sin\theta_a; \cos\theta_a; 0)^T$  and  $X_{a\varphi} = (\cos\theta_a \cos\varphi_a; \sin\theta_a \cos\varphi_a; -\sin\varphi_a)^T$ . Now,  $U_{a\theta}$  and  $U_{a\varphi}$  are the inputs that are written in terms of agent  $a$  and are given as;

$$\begin{cases} U_{a\theta} = -\sum_{b \in n_a} \sin(\varphi_b) \sin(\theta_a - \theta_b) \\ U_{a\varphi} = -\sum_{b \in n_a} \sin(\varphi_a) \cos(\varphi_b) - \sin(\varphi_b) \cos(\varphi_a) \cos(\theta_a - \theta_b) \end{cases} \quad (8)$$

The above inputs were developed for the arrangement of velocity vectors of a group of kinematic agents.

#### 4. Designed Algorithm

The design approach for this study will be to use the PSO algorithm to find the best agent and then implement Vicsek MAS for synchronization and connectivity between the two different clusters. This strategy is explained in detail below:

##### 4.1. Particle Swarm Optimization

PSO is an optimization method inspired by the behavior of a flock of birds or a school of fish. It starts with each particle having a random position and velocity. Then, it continuously updates the position and velocity of each particle until the global best solution is found [33,34]. Particles revise their positions according to the previous value, their individual best values, and the global best value. Their new positions mainly depend on their previous positions and the present velocities [35]. If the current position is better than the earlier positions, it turns into the individual best position for that particle. If the current position of the particle is better than all the particles' positions, then it is set as the global best value [36,37]. Given the route planning is in a three-dimensional plane and each particle's waypoint is  $S$ , so for the  $i$ th particle, the position vector  $P_i$  and the velocity vector  $V_i$  can be given as:

$$\begin{cases} P_i = [P_{(i,1)}, \dots, P_{(i,S)}]^T = [(P^x, P^y, P^z)_{(i,1)}, \dots, (P^x, P^y, P^z)_{(i,S)}]^T \\ V_i = [V_{(i,1)}, \dots, V_{(i,S)}]^T = [(V^x, V^y, V^z)_{(i,1)}, \dots, (V^x, V^y, V^z)_{(i,S)}]^T \end{cases} \quad (9)$$

In the above equation,  $j \in 1, \dots, S$ ;  $(P^x, P^y, P^z)_{(i,j)}$  and  $(V^x, V^y, V^z)_{(i,j)}$  denote the position and velocity of the  $j$ th waypoint for the  $i$ th particle in a 3-dimensional plane.

Given that  $D$  denotes the total number of particles, the swarm can be represented as:

$$[(P_1, V_1), (P_2, V_2), \dots, (P_D, V_D)] \quad (10)$$

For a swarm containing  $D$  particles, there are  $D$  local best and global best solutions:

$$P_{(i,best)} = [P_{(i,1,best)}, \dots, P_{(i,S,best)}]^T = [(P^x, P^y, P^z)_{(i,1,best)}, \dots, (P^x, P^y, P^z)_{(i,S,best)}]^T \quad (11)$$

$$P_{best}^G = [P_{1,best}, \dots, P_{S,best}]^T = [(P^x, P^y, P^z)_{(1,best)}, \dots, (P^x, P^y, P^z)_{(S,best)}]^T \quad (12)$$

In the above equation,  $i \in 1, \dots, D$ . The multiple objective PSO function is given as:

$$P_{(i,best)}(t+1) = \begin{cases} P_{(i,best)}(t); \text{ if } f(P_{(i,best)}(t)) \leq f(P_i(t+1)) \\ P_i(t+1); \text{ if } f(P_{(i,best)}(t)) > f(P_i(t+1)) \end{cases} \quad (13)$$

$$P_{best}^G \in (P_{(i,1,best)}, \dots, P_{(i,S,best)}) \quad (14)$$

$$f(P_{best}^G(t)) = \min[f(P_{best,1}(t)), \dots, f(P_{best,S}(t))] \quad (15)$$

The position and velocity of the particles in the swarm update according to the following equations :

$$\begin{cases} P_{i,j}(t+1) = P_{i,j}(t) + V_{i,j}(t+1) \\ V_{i,j}(t+1) = w_i \times V_{(i,j)}(t) + a_1 \times r_1 \times (P_{(i,j,best)}(t) - P_{(i,j)}(t)) + a_2 \times r_2 \times (P_{(G,j)}^{best}(t) - P_{(i,j)}(t)) \end{cases} \quad (16)$$

In the above equation, coefficients of acceleration are  $a_1$  and  $a_2$ , while  $r_1$  and  $r_2$  represent any number from 0 to 1. Also,  $w_i$  represents the inertial weight.

By changing  $w_i$ , the performance of the algorithm can be fine-tuned. It can help in stabilizing the individual and global best results of each particle. For a better global outcome, increase  $w_i$ , and to enhance the local results, decrease it. The inertial weight is given as:

$$w_i = \frac{(w_{imax} - w_{imin})t_c}{T_m} \quad (17)$$

In the above equation,  $t_c$  represents the current instance of the PSO,  $T_m$  denotes the highest number of iterations, and  $w_{imax}$  and  $w_{imin}$  present the highest and lowest value of  $w_i$ .

#### 4.2. Vicsek Model

The Vicsek MAS model consists of  $n$  independent agents traveling with identical speeds through the model. It revises the direction of each agent according to the heading of its neighbor agent. The neighboring agents of agent  $a$  ( $1 \leq a \leq n$ ) at any time instance  $t$  are within a radius  $r$  centered at agent  $a$  and are denoted as  $N_a(t)$ ,

$$\begin{cases} N_a(t) = \{b | d_{ab}(t) < r\} \\ d_{ab}(t) = \sqrt{(x_a(t) - x_b(t))^2 + (y_a(t) - y_b(t))^2} \end{cases} \quad (18)$$

In the above equation, the coordinates of agent  $a$  are  $(x_a(t), y_a(t))$  at time  $t$ . The neighboring agent of agent  $a$  is agent  $b$ . All the agents in the model travel with the same velocity  $v$ .

$$\begin{cases} x_a(t+1) = x_a(t) + v \cos \theta_a(t) \\ y_a(t+1) = y_a(t) + v \sin \theta_a(t) \\ \theta_a(t+1) = \arctan \frac{\sum_{b \in N_a(t)} \sin \theta_b(t)}{\sum_{b \in N_a(t)} \cos \theta_b(t)} \end{cases} \quad (19)$$

In the above equation, the heading angle of agent  $a$  is given as  $\theta_a(t)$ . The Vicsek MAS is a dynamic model, and this paper uses basic graph theory (discussed in Section 3) to examine this system. The equation for  $\theta_a(t)$  can be rewritten as,

$$\tan \theta_a(t+1) = \sum_{b \in N_a(t)} \frac{\cos \theta_b(t)}{\sum_{k \in N_a(t)} \cos \theta_k(t)} \tan \theta_a(t) \quad (20)$$

$$\tan \theta(t+1) = A(t) \tan \theta(t) \quad (21)$$

In the above equation,  $\tan \theta(t) \triangleq (\tan \theta_1(t), \dots, \tan \theta_N(t))^T$ ,  $A(t)$  is the average weighted matrix of the graph  $\hat{G}_t$  and is given as;

$$\hat{a}_{ab}(t) = \begin{cases} \frac{\cos \theta_b(t)}{\sum_{k \in N_a(t)} \cos \theta_k(t)} & \text{if } (a, b) \in \mathcal{E}_t \\ 0, & \text{otherwise} \end{cases} \quad (22)$$

To analyze the synchronizing behavior of the Vicsek model, the linear version of the  $\theta_a(t)$  can be written as:

$$\theta_a(t+1) = \frac{1}{n_a(t)} \sum_{b \in N_a(t)} \theta_b(t) \quad (23)$$



In the above equation,  $N_a(t)$  contains  $n_a(t)$  elements. Likewise, Equation (22) can be rewritten as;

$$\tan \theta(t+1) = \tilde{A}(t)\theta(t) \quad (24)$$

In the above equation,  $\theta(t) \triangleq (\theta_1(t), \dots, \theta_N(t))^T$ , and the elements of the matrix  $\tilde{A}(t)$  can be given as

$$\tilde{a}_{ab}(t) = \begin{cases} \frac{1}{n_a(t)}, & \text{if } (a,b) \in \mathcal{E}_t \\ 0, & \text{otherwise} \end{cases} \quad (25)$$

#### 4.3. Synchronization and Connectivity

To better understand the synchronization and connectivity of the Vicsek model, first, this study must clearly describe synchronization. The above-mentioned Vicsek model is said to synchronize when the heading angles of all the agents satisfy the following equation:

$$\lim_{t \rightarrow \infty} \theta_a(t) = \theta, \quad a = 1, \dots, N \quad (26)$$

In the above equation,  $\theta$  depends on the initial values  $\{\theta_a(0), x_a(0), y_a(0), a = 1, \dots, N\}$  and the parameters of  $r$  and  $v$ . The propositions given below will establish the synchronization for the Vicsek model and its linearized version.

Considering the Vicsek model of Equation (20), suppose that the starting neighbor graph  $\hat{G}_0 = \{\mathcal{V}, \mathcal{E}_0\}$  is connected and  $\{\theta_a(0) \in (-\frac{\pi}{2}, \frac{\pi}{2}), a = 1, \dots, N\}$ , therefore, the Vicsek model will synchronize if it satisfies the condition:

$$\left\{ \begin{array}{l} v \leq \frac{d}{\Delta_0} \left( \frac{\cos \bar{\theta}}{N} \right)^N \\ \bar{\theta} = \max_i |\theta_a(0)| \\ d = r - \max_{a,b \in \mathcal{E}_0} d_{ab}(0) \\ \Delta_0 = \max_{a,b} \{ \tan \theta_a(0) - \tan \theta_b(0) \} \end{array} \right. \quad (27)$$

In the above equation,  $N$  represents the number of total agents. Considering the linear version of the model in (20) and (24), let the starting graph  $\hat{G}_0$  be connected and  $\theta_a(0) \in [0, 2\pi)$ . Therefore, the Vicsek model will synchronize if it satisfies the condition:

$$v \leq \frac{d \left( \frac{1}{N} \right)^N}{2\pi} \quad (28)$$

#### 4.4. Dynamic Leader Selection

The formation configuration sometimes undergoes variations in systems with multiple agents because of communication failures among agents. To model a random communication failure, each link  $(a,b) \in E$  individually fails with  $p$  probability. Let the graph topology for this communication failure model be  $\hat{G}$  and  $E_{\hat{G}}(t_{conv})$  be the estimated convergence time. While  $E_{\hat{G}}(\cdot)$  represents the expectation of the argument over the group of network formations, denoted by  $\hat{G}$ . Decreasing  $E_{\hat{G}}(t_{conv})$  will maximize the convergence rate. Therefore, the formula to select a leader  $k$  for maximizing the convergence rate is,

$$\max_Y E_{\hat{G}} \left( \min_{x(0)} x(0)^T (Y\hat{L} + \hat{L}Y)x(0) \right) \quad (29)$$

Such that it satisfies the following conditions,

$$\left\{ \begin{array}{l} \text{tr}(Y) \geq n - k \\ Y_{aa} \in \{0, 1\} \forall_a \neq V \\ Y_{ab} = 0 \forall_a \neq b \end{array} \right. \quad (30)$$

The objective function is consistent with the expected convergence rate over the potential network formations. As  $\min_{x(0)} x(0)^T (Y\hat{L} + \hat{L}Y)x(0)$  is a convex function of  $Y$ , which is the MAS convergence rate, the objective function of (30) is also convex because it is the weighted sum of convex functions.

#### 4.5. B-Spline Path Smoothing

The path planned by the hybrid algorithm mostly consists of a series of line segments. To make sure that the path generated is smooth and flightworthy, the B-spline curve strategy is used. The B-spline curve is an improvement of the Bezier curve method and has the advantages of preserving convexity and geometrical invariability.

Mathematically, the B-spline curve can be given as [38];

$$P(u) = \sum_{a=0}^n d_a N_{a,b}(u) \quad (31)$$

In the above equation,  $d_a (i = 0, 1, \dots, n)$  are the controlling points, and  $N_{a,b}(u)$  are the normalized b-order B-spline functions and are defined with the help of the Cox-de Boor recursion method as;

$$\begin{cases} N_{a,b}(u) = \begin{cases} 1, & \text{if } u_a \leq u \leq u_{a+1} \\ 0, & \text{otherwise} \end{cases} \\ N_{a,b}(u) = \frac{u-u_a}{u_{a+b}-u_a} N_{a,b-1}(u) + \frac{u_{a+b+1}-u}{u_{a+b+1}-u_{a+1}} N_{a+1,b-1}(u) \\ \text{define } N_{0,0} = 0 \end{cases} \quad (32)$$

The parametric knots  $\{u_0 \leq u_1 \leq \dots \leq u_{n+b}\}$  determine the basic functions of the B-spline curve. The B-spline curve is not affected by just moving a single control point, as opposed to the Bezier curve. Another advantage over the Bezier curve is that increasing the control points will not increase the degree of polynomials.

## 5. Flowchart and Algorithm

This section details the flowchart and the algorithm of the proposed method. Figure 5 presents the flowchart of the designed strategy. As it is clear from the flowchart, the algorithm starts by initializing the parameters and setting the starting and ending points for the mission. Afterward, the algorithm performs the cluster allocation. Then, it sets the particles of cluster 1 and 2 at random positions and starts updating their position and velocity until it finds the global best solution. Later, MAS designates that particle found in the global best solution as the leader of that cluster, and the rest of the particles start to follow the leader as agents in a formation.

Moreover, the Vicsek model ensures synchronization and connectivity within each cluster and the larger swarm as well. Finally, the strategy uses dynamic leader selection to select the leader of the swarm and determines if there is any obstacle in the path of the swarm. If there is an obstacle, the algorithm updates the heading angles of the agents and selects a different leader for the swarm dynamically according to the mission requirement. The strategy repeats this process until the destination is found.

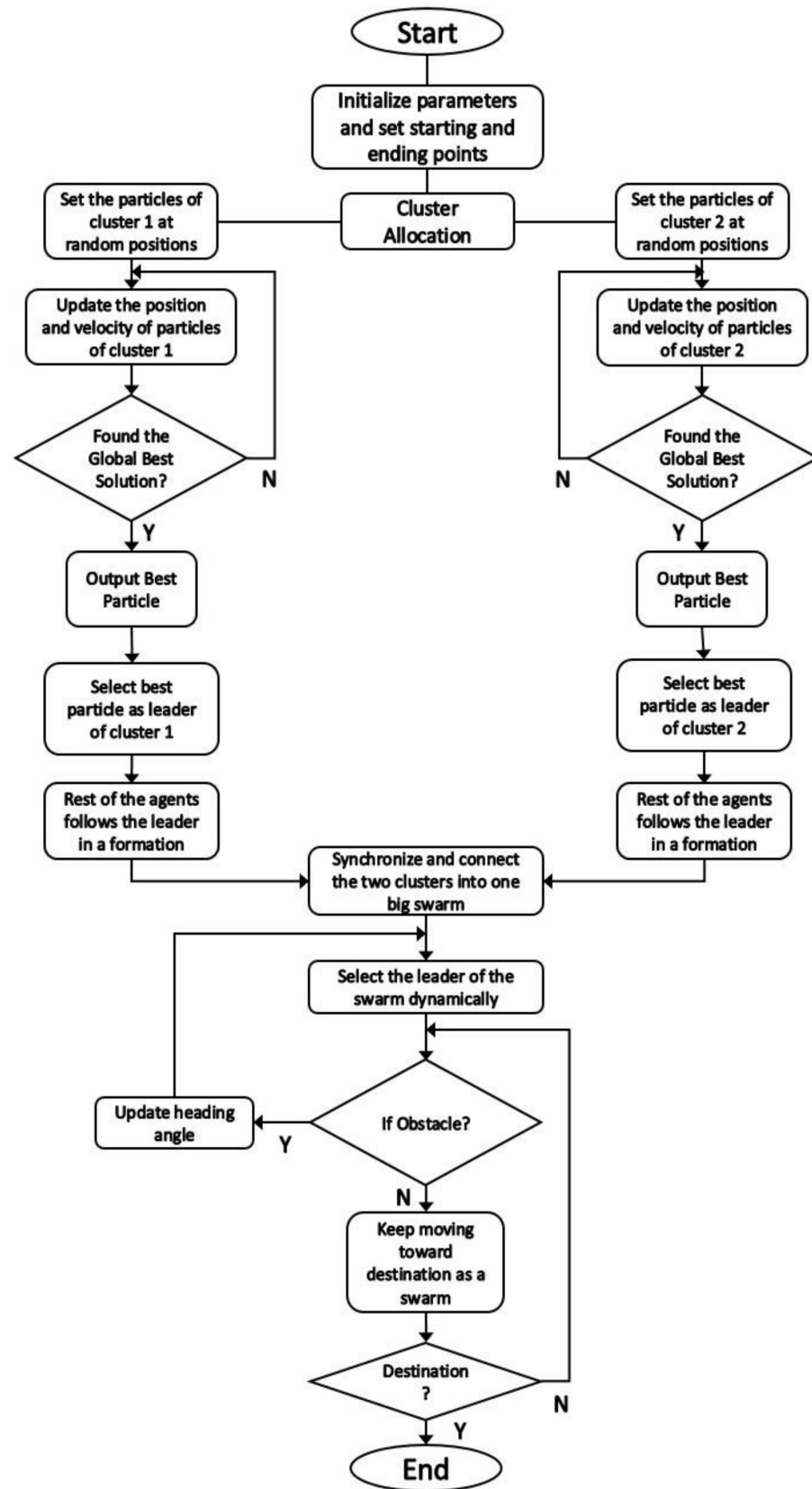


Figure 5. Flowchart of the Proposed Method.

The Algorithm 1 of the proposed method is detailed below:

---

**Algorithm 1** The pseudo code

---

```

Initialize the parameters and set terrain using Equation (a); //Algorithm Initialization
Set starting and ending points for the swarm;
Allocate clusters using Equation (1);
Set UAVNumber as 10, ClusterNumber as 2, and ClusterSize as 5;
Set  $T_m$  as 50; //  $T_m$  denotes number of iterations
  while ( $a < T_m$ )
    while ( $b < \text{ClusterNumber}$ )
      while ( $c < \text{ClusterSize}$ ) //Finding best particle
        Update swarm particles using Equation (16);
        Calculate  $P_{best}$  and  $P_{best}^G$  using Equations (11) and (12);
        Replace the previous iteration of  $P_{best}^G$  if better in the current iteration;
      end while
      return best particle //Return best particle of the cluster
    while ( $c < \text{ClusterSize}$ ) //Multi-agent System
      Update the position and heading of agents using Equation (19);
      The model synchronizes if satisfies Equation (27); //Synchronization
    end while
  end while
  while ( $b < \text{UAVNumber}$ ) //Dynamic leader selection
    Select a leader dynamically using Equations (29) and (30);
  end while
  return leader
end while

```

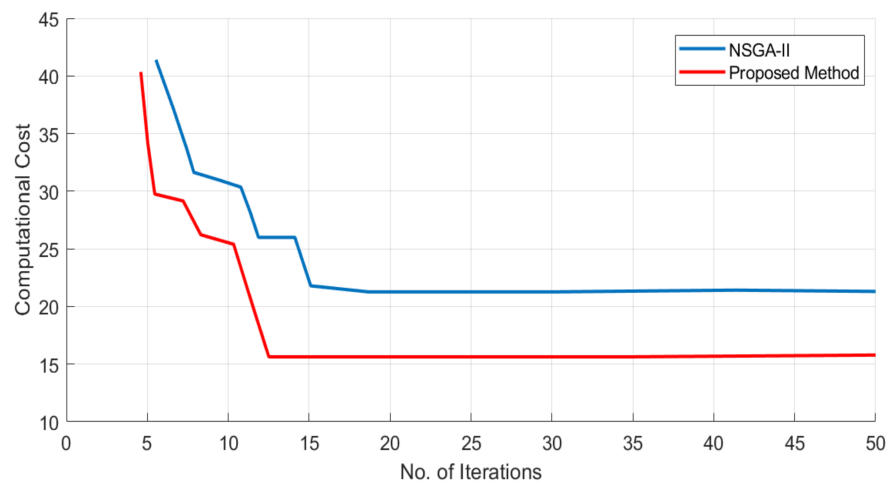
---

## 6. Simulations

This section proves the efficacy of the proposed hybrid algorithm using simulations. The mission area is 30 km long, 35 km wide, and the altitude is 3 km. First, this study compares the performance of the proposed method with the NSGA-II genetic algorithm. Then, the study will implement the designed algorithm in two different scenarios. The first scenario contains two clusters of five UAVs each at random positions. The task is to bring these UAVs into the desired formation using PSO and MAS. The second task concerns joining the two clusters to form one big swarm. The synchronization and connectivity between the UAVs are ensured by the Vicsek MAS model.

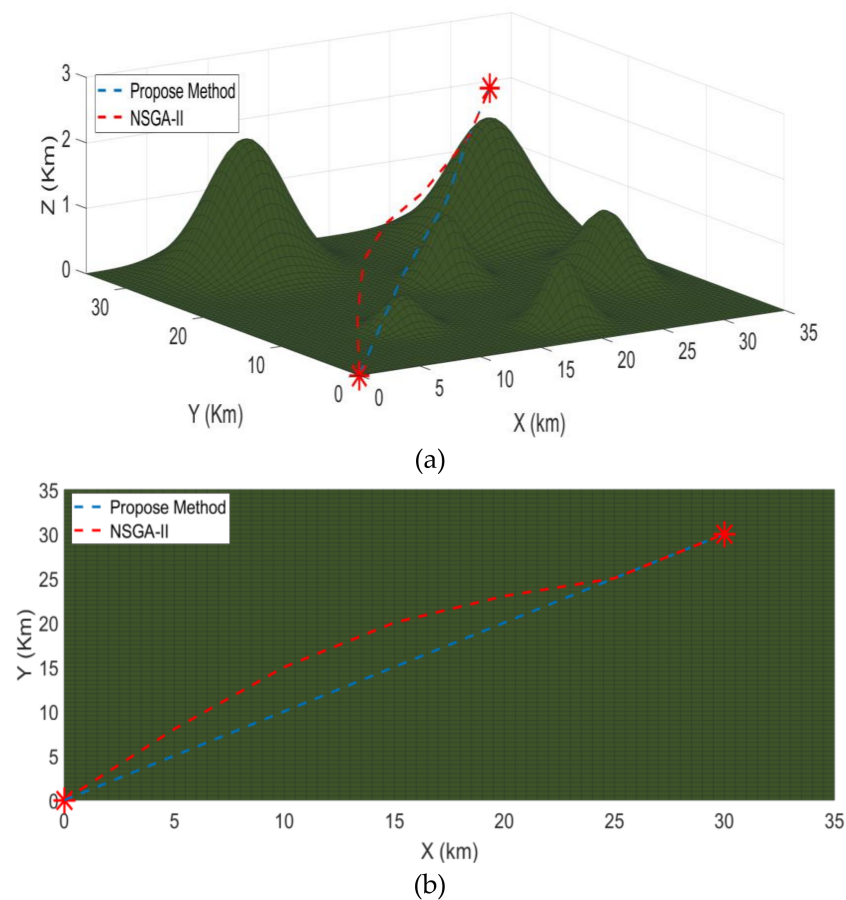
### 6.1. Comparison with NSGA-II

To test the efficiency of the designed strategy, this paper first compares it with the genetic algorithm NSGA-II. Figure 6 shows the comparison between the computational cost of the NSGA-II and the proposed method. As it is clear from the figure, the proposed method requires less computational power as compared to NSGA-II. Also, the proposed method takes fewer iterations to find the optimal solution.

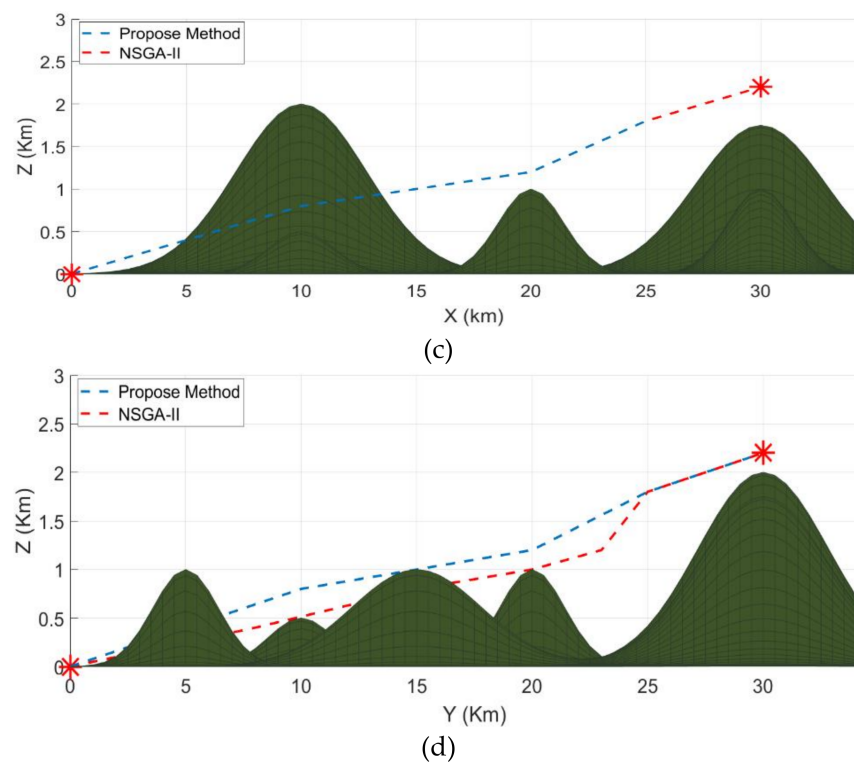


**Figure 6.** Performance comparison between the proposed method and the NSGA-II; the  $y$ -axis represents the computational cost, and the  $x$ -axis shows the number of iterations.

Figure 7 presents the comparison between the two algorithms from a different perspective. Figure 7a shows the 3D version while Figure 7b–d represent the XY, XZ, and YZ axis versions, respectively. It is clear from the figure that the proposed method is more optimal than the NSGA-II. It takes less time and distance to arrive at the destination while still avoiding the obstacles. It is straighter and takes less turns, which help in remaining stable during the flight.



**Figure 7.** Cont.



**Figure 7.** Comparison between NSGA-II and the proposed method; (a) shows the 3D version while (b–d) represent the XY, XZ, and YZ axis versions, respectively. The legend is the same for all simulations.

### 6.2. Problem Case 1

The first scenario deals with organizing the two clusters into formations and then maintaining that formation for the second scenario. In the first scenario, 10 fixed-wing UAVs are divided into two clusters and each cluster has a leader and four followers. It means five UAVs in each cluster work collaboratively to perform a task. The terrain model contains different obstacles like mountains. The aerial vehicles have a challenging time avoiding the hazardous peaks, maintaining their strategy, and reaching the designated area safely. Hence, it is necessary to maintain the desired synchronization and connectivity with their neighbor agents. Figure 8 presents different perspectives of the first scenario. The Figure 8a shows the 3D version while Figure 8b–d represent the XZ, XY, and YZ axis versions, respectively. First, PSO finds the best particle in each cluster, and then the MAS designates that particle as the leader of that cluster. Afterward, the rest of the UAVs start to follow in a leader-follower formation. As it is clear in the simulation, the two clusters are now in the leader-follower formation with two leaders each having four followers. The two clusters maintain their formation even through the mountainous terrain and narrow spaces, as evident from the top-right simulation. The proposed method performed the task perfectly and the two clusters are now ready to be synchronized into one big cluster in the next scenario.

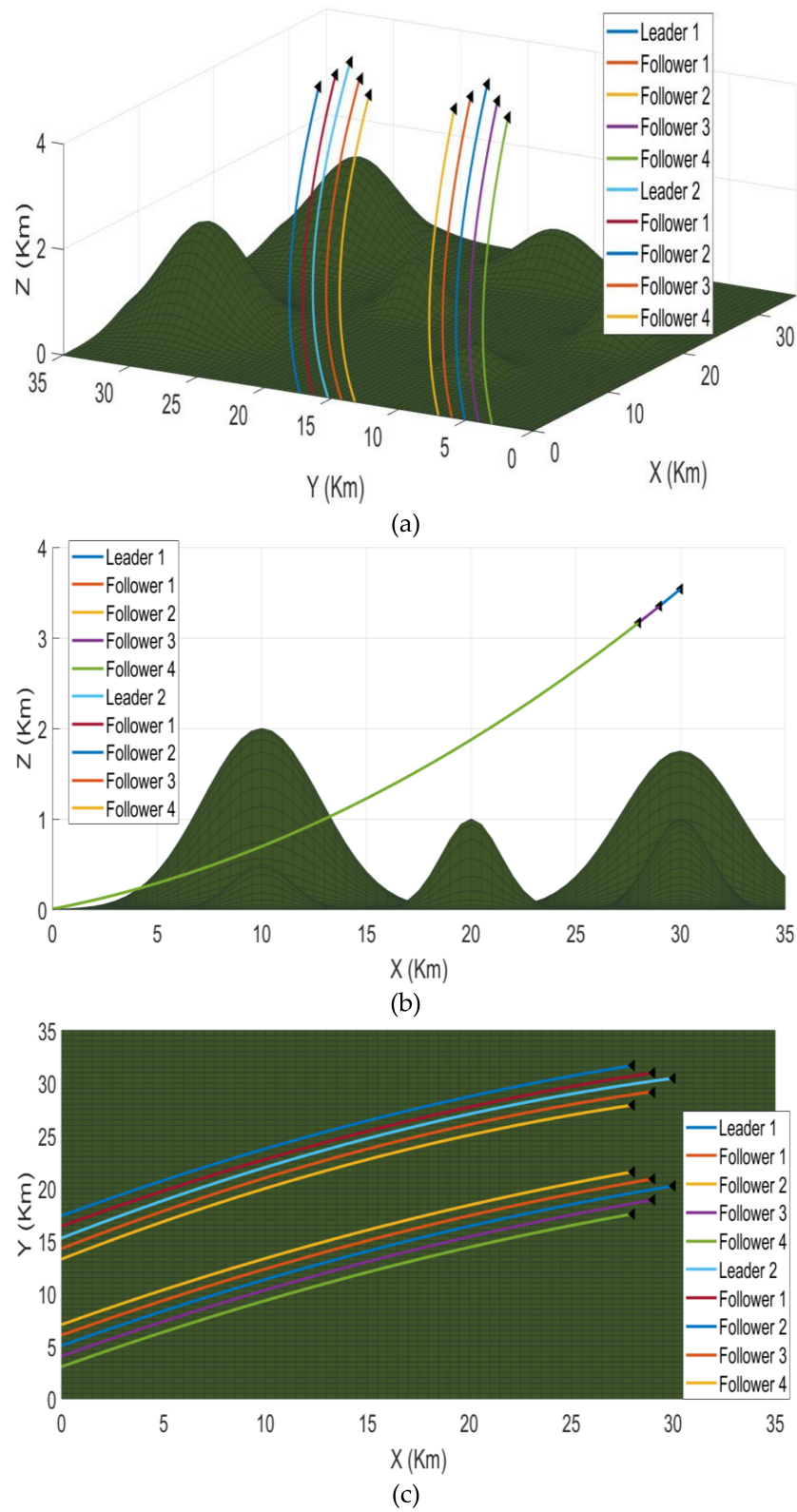
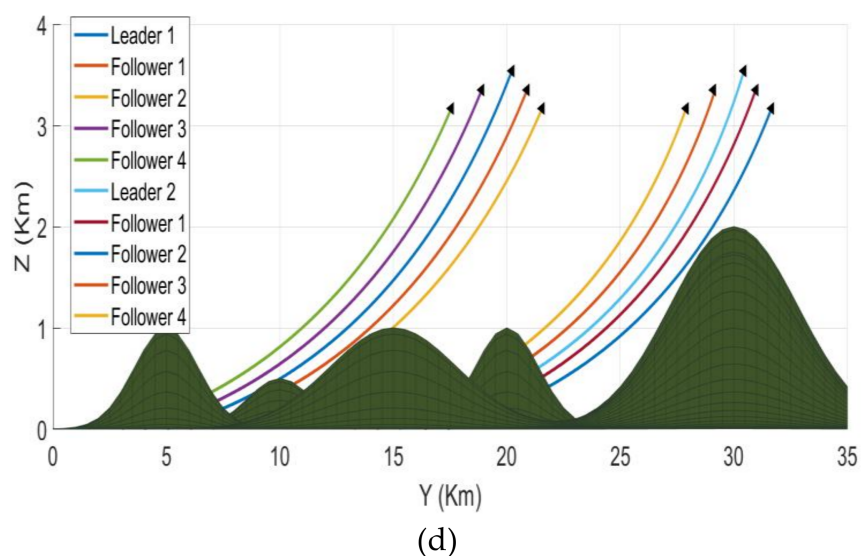


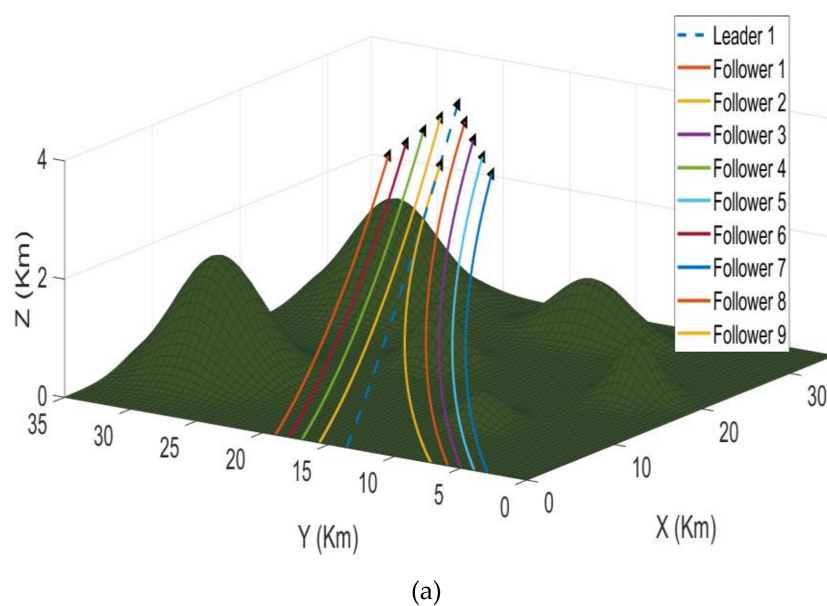
Figure 8. Cont.



**Figure 8.** The first mission scenario containing two clusters with five UAVs each reaching a leader-follower formation; (a) shows the 3D version while (b–d) represent the XZ, XY, and YZ axis versions, respectively. The legend is the same for all simulations.

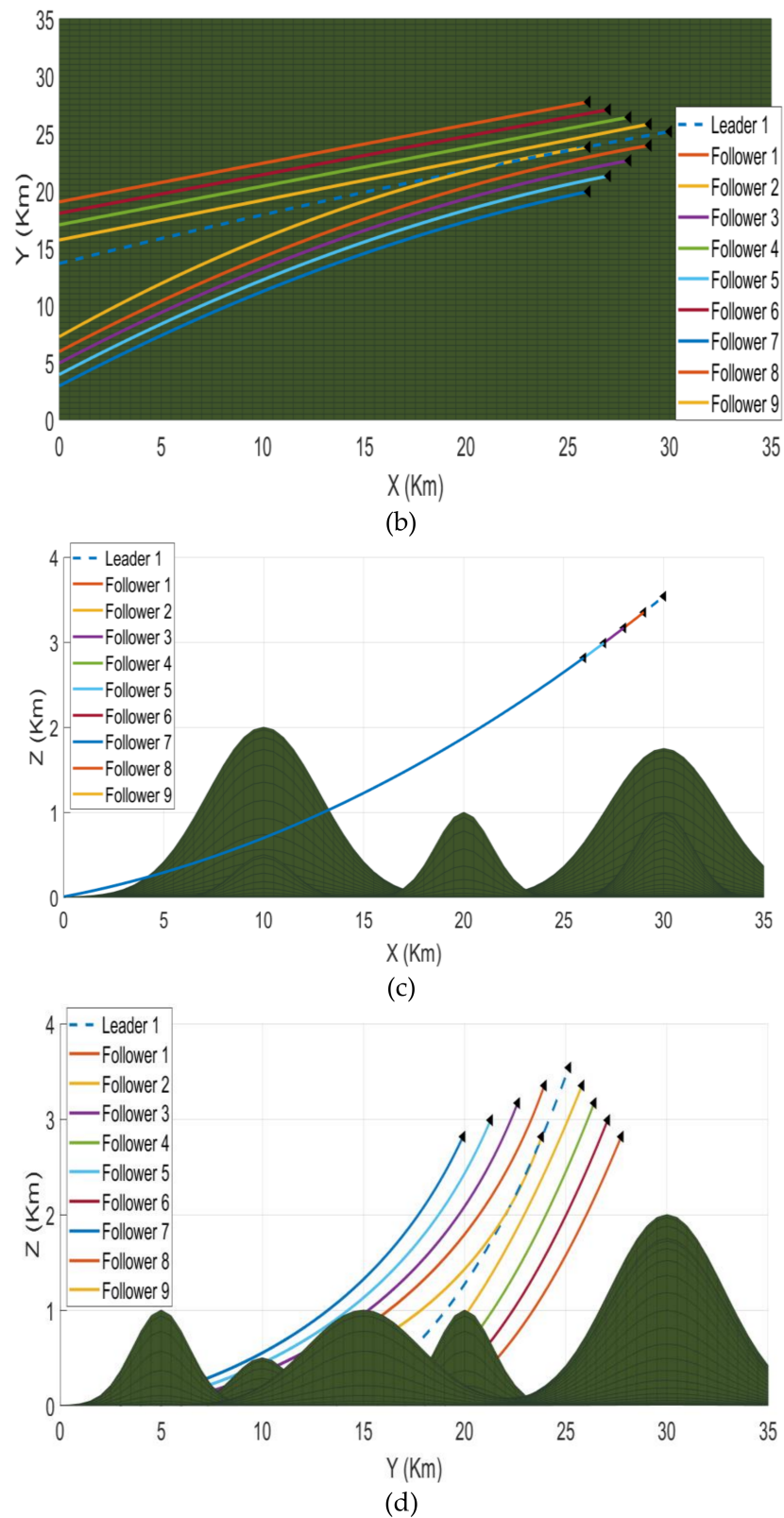
### 6.3. Problem Case 2

The second scenario starts where the first scenario ended. The second scenario starts after successfully maintaining the formation of both the clusters and fulfilling their separate mission tasks in the first scenario. Afterward, both the clusters join in a common cluster called a swarm. At that instant of time, perfect coordination or communication between both the clusters is required to choose which leader will lead the formation of a swarm now. To help achieve the above objective, the algorithm will dynamically select the leader. Figure 9 presents different perspectives of the second scenario in simulation. The Figure 9a shows the 3D version while Figure 9b–d represent the XY, XZ, and YZ axis versions, respectively. As it is clear, the algorithm has transformed the two clusters into one swarm with the leader being selected dynamically.



**Figure 9.** Cont.





**Figure 9.** The second mission scenario where the two clusters merge into one big swarm; (a) shows the 3D version while (b–d) represent the XY, XZ, and YZ axis versions, respectively. The legend is the same for all simulations.

## 7. Conclusions

This study proposed a collective motion and self-organization control of a swarm of 10 UAVs, which were divided into two clusters. This article designed a hybrid meta-heuristic algorithm by merging the particle swarm optimization (PSO) with the multi-agent system (MAS). PSO provided the best agents of a cluster. Afterward, the MAS assigned the best agent as the leader of the cluster. Moreover, the leader found the optimal path for each cluster. Initially, each cluster had agents at random positions. Later, the clusters were organized into a formation by implementing PSO with the MAS model. This helped coordinate the agents inside the cluster. However, for the connectivity and synchronization of the common cluster, this study used the Vicsek model along with dynamic leader selection ability. The flying formations of each cluster and the swarm were created by the B-spline curve based on simple waypoint-defined graph theory. This paper compared the designed algorithm with the NSGA-II to show that the proposed model has better convergence and durability both inside the clusters and inside the greater swarm. Lastly, the simulations showed that the algorithm is effective and completed both the scenarios according to the mission requirements.

The proposed algorithm can be applied to any number of clusters with any number of UAVs. However, this study focused on a specific scenario of 10 UAVs with two clusters. For future plans, the authors plan to apply the proposed method to more clusters and with more UAVs than 10. The authors also intend to compare the efficiency of the proposed method with other state-of-the-art algorithms in the near future. The authors also plan to implement the proposed algorithm on hardware and compare the experimental results with the simulation results.

**Author Contributions:** The overall Concept, methodology and software validation is done by Z.A.A.; Z.H. confirms the validation, Z.A.A. and R.J.M. investigate and writing—original draft preparation, Z.A.A. and Z.H.; writing—review and editing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This above research work supported by the Startup fund of Zain, School of Systems Science, Beijing Normal University, Zhuhai and Natural Science Foundation under grant number 61374165.

**Data Availability Statement:** All the data used to support the findings of this study are included within the article.

**Conflicts of Interest:** There is no conflict of interest between all the authors of this article.

## References

1. Zhang, B.; Sun, X.; Liu, S.; Deng, X. Tracking control of multiple unmanned aerial vehicles incorporating disturbance observer and model predictive approach. *Trans. Inst. Meas. Control* **2019**, *42*, 951–964. [[CrossRef](#)]
2. Cai, Z.; Wang, L.; Zhao, J.; Wu, K.; Wang, Y. Virtual target guidance-based distributed model predictive control for formation control of multiple UAVs. *Chin. J. Aeronaut.* **2020**, *33*, 1037–1056. [[CrossRef](#)]
3. Das, P.K.; Jena, P.K. Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators. *Appl. Soft Comput.* **2020**, *92*, 106312. [[CrossRef](#)]
4. Fan, X.; Sayers, W.; Zhang, S.; Han, Z.; Ren, L.; Chizari, H. Review and Classification of Bio-inspired Algorithms and Their Applications. *J. Bionic Eng.* **2020**, *17*, 611–631. [[CrossRef](#)]
5. Ji, B.; Lu, X.; Sun, G.; Zhang, W.; Li, J.; Xiao, Y. Bio-Inspired Feature Selection: An Improved Binary Particle Swarm Optimization Approach. *IEEE Access* **2020**, *8*, 85989–86002. [[CrossRef](#)]
6. Qiu, H.; Duan, H. A multi-objective pigeon-inspired optimization approach to UAV distributed flocking among obstacles. *Inf. Sci.* **2020**, *509*, 515–529. [[CrossRef](#)]
7. Yang, Q.; Yoo, S.-J. Optimal UAV path planning: Sensing data acquisition over IoT sensor networks using multi-objective bio-inspired algorithms. *IEEE Access* **2018**, *6*, 13671–13684. [[CrossRef](#)]
8. Duan, H.; Li, P. *Bio-Inspired Computation in Unmanned Aerial Vehicles*; Springer: Berlin/Heidelberg, Germany, 2014.
9. Xin, J.; Zhong, J.; Yang, F.; Cui, Y.; Sheng, J. An Improved Genetic Algorithm for Path-Planning of Unmanned Surface Vehicle. *Sensors* **2019**, *19*, 2640. [[CrossRef](#)]
10. Ghamry, K.A.; Kamel, M.A.; Zhang, Y. Multiple UAVs in forest fire fighting mission using particle swarm optimization. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 1404–1409.

11. Teng, H.; Ahmad, I.; Msm, A.; Chang, K. 3D Optimal Surveillance Trajectory Planning for Multiple UAVs by Using Particle Swarm Optimization with Surveillance Area Priority. *IEEE Access* **2020**, *8*, 86316–86327. [[CrossRef](#)]
12. Shao, Z.; Yan, F.; Zhou, Z.; Zhu, X. Path planning for multi-UAV formation rendez-vous based on distributed cooperative particle swarm optimization. *Appl. Sci.* **2019**, *9*, 2621. [[CrossRef](#)]
13. Shao, S.; Peng, Y.; He, C.; Du, Y. Efficient path planning for UAV formation via comprehensively improved particle swarm optimization. *ISA Trans.* **2020**, *97*, 415–430. [[CrossRef](#)]
14. Khan, M.W.; Wang, J. The research on multi-agent system for microgrid control and optimization. *Renew. Sustain. Energy Rev.* **2017**, *80*, 1399–1411. [[CrossRef](#)]
15. Mu, B.; Zhang, K.; Shi, Y. Integral Sliding Mode Flight Controller Design for a Quadrotor and the Application in a Heterogeneous Multi-Agent System. *IEEE Trans. Ind. Electron.* **2017**, *64*, 9389–9398. [[CrossRef](#)]
16. Wang, Y.; Cheng, Z.; Xiao, M. UAVs' formation keeping control based on Multi-Agent system consensus. *IEEE Access* **2020**, *8*, 49000–49012. [[CrossRef](#)]
17. Shirani, B.; Najafi, M.; Izadi, I. Cooperative load transportation using multiple UAVs. *Aerosp. Sci. Technol.* **2019**, *84*, 158–169. [[CrossRef](#)]
18. Sharma, A.; Bahl, S.; Bagha, A.K.; Javaid, M.; Shukla, D.K.; Haleem, A. Multi-agent system applications to fight COVID-19 pandemic. *Apollo Med.* **2020**, *17*. [[CrossRef](#)]
19. Dai, Q.; Liu, J.; Wei, Q. Optimal photovoltaic/battery energy storage/electric vehicle charging station design based on multi-agent particle swarm optimization algorithm. *Sustainability* **2019**, *11*, 1973. [[CrossRef](#)]
20. Roshanzamir, M.; Balafar, M.A.; Razavi, S.N. A new hierarchical multi group particle swarm optimization with different task allocations inspired by holonic multi agent systems. *Expert Syst. Appl.* **2020**, *149*, 113292. [[CrossRef](#)]
21. Biswas, S.; Anavatti, S.G.; Garratt, M.A. Obstacle avoidance for multi-agent path planning based on vectorized particle swarm optimization. In *Intelligent and Evolutionary Systems*; Springer: Cham, Switzerland, 2017; pp. 61–74.
22. Kesavamoorthy, R.; Soundar, K.R. Swarm intelligence based autonomous DDoS attack detection and defense using multi agent system. *Clust. Comput.* **2019**, *22*, 9469–9476. [[CrossRef](#)]
23. Ansari, S.; Ahmad, J.; Shah, S.A.; Bashir, A.K.; Boutaleb, T.; Sinanovic, S. Chaos-based privacy preserving vehicle safety protocol for 5G Connected Autonomous Vehicle networks. *Trans. Emerg. Telecommun. Technol.* **2020**, *31*. [[CrossRef](#)]
24. Vicsek, T.; Czirók, A.; Ben-Jacob, E.; Cohen, I.; Shochet, O. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.* **1995**, *75*, 1226. [[CrossRef](#)]
25. Xiao, Y.; Song, C.; Tian, L.; Liu, Y.-Y. Accelerating the Emergence of Order in Swarming Systems. *Adv. Complex Syst.* **2019**, *23*. [[CrossRef](#)]
26. Leitão, P.; Barbosa, J.; Trentesaux, D. Bio-inspired multi-agent systems for reconfigurable manufacturing systems. *Eng. Appl. Artif. Intell.* **2012**, *25*, 934–944. [[CrossRef](#)]
27. Zambonelli, F.; Omicini, A.; Anzenberger, B.; Castelli, G.; de Angelis, F.L.; Serugendo, G.d.M.; Dobson, S.; Fernandez-Marquez, J.L.; Ferscha, A.; Mamei, M.; et al. Developing pervasive multi-agent systems with nature-inspired coordination. *Pervasive Mob. Comput.* **2015**, *17*, 236–252. [[CrossRef](#)]
28. Rodriguez-Angeles, A.; Vazquez Chavez, L.F. Bio-inspired decentralized autonomous robot mobile navigation control for multi agent systems. *Kybernetika* **2018**, *54*, 135–154.
29. Skobelev, P.; Budaev, D.; Gusev, N.; Voschuk, G. Designing Multi-agent Swarm of UAV for Precise Agriculture. In *Highlights of Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection, Proceedings of the International Workshops of PAAMS 2018, Toledo, Spain, 20–22 June 2018*; Bajo, J., Ed.; Springer: Cham, Switzerland, 2018; Volume 887, pp. 47–59. [[CrossRef](#)]
30. Moshtagh, N.; Jadbabaie, A.; Daniilidis, K. Vision-based Distributed Coordination and Flocking of Multi-agent Systems. *Math. Phys.* **2005**, *16*, 31. [[CrossRef](#)]
31. Clark, A.; Alomair, B.; Bushnell, L.; Poovendran, R. Leader selection in multi-agent systems for smooth convergence via fast mixing. In Proceedings of the 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), Maui, HI, USA, 10–13 December 2012; pp. 818–824.
32. Hu, X.; Ma, H.; Ye, Q.; Luo, H. Hierarchical method of task assignment for multiple cooperating UAV teams. *J. Syst. Eng. Electron.* **2015**, *26*, 1000–1009. [[CrossRef](#)]
33. Liu, Y.; Zhang, X.; Guan, X.; Delahaye, D. Adaptive sensitivity decision based path planning algorithm for unmanned aerial vehicle with improved particle swarm optimization. *Aerosp. Sci. Technol.* **2016**, *58*, 92–102. [[CrossRef](#)]
34. Cheng, R.; Jin, Y. A social learning particle swarm optimization algorithm for scalable optimization. *Inf. Sci.* **2015**, *291*, 43–60. [[CrossRef](#)]
35. Ang, K.M.; Lim, W.H.; Isa, N.A.M.; Tiang, S.S.; Wong, C.H. A constrained multi-swarm particle swarm optimization without velocity for constrained optimization problems. *Expert Syst. Appl.* **2020**, *140*, 112882. [[CrossRef](#)]
36. Selvakumar, A.I.; Thanushkodi, K. A new particle swarm optimization solution to nonconvex economic dispatch problems. *IEEE Trans. Power Syst.* **2007**, *22*, 42–51. [[CrossRef](#)]
37. Taghiyeh, S.; Xu, J. A new particle swarm optimization algorithm for noisy optimization problems. *Swarm Intell.* **2016**, *10*, 161–192. [[CrossRef](#)]
38. Qu, C.; Gai, W.; Zhong, M.; Zhang, J. A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning. *Appl. Soft Comput.* **2020**, *89*, 106099. [[CrossRef](#)]