

Article

# A Multi-Colony Social Learning Approach for the Self-Organization of a Swarm of UAVs

Muhammad Shafiq<sup>1</sup>, Zain Anwar Ali<sup>1,\*</sup>, Amber Israr<sup>1</sup>, Eman H. Alkhamash<sup>2</sup> and Myriam Hadjouni<sup>3</sup> 

<sup>1</sup> Electronic Engineering Department, Sir Syed University of Engineering & Technology, Karachi 75300, Pakistan; muhshafiq@ssuet.edu.pk (M.S.); aisrar@ssuet.edu.pk (A.I.)

<sup>2</sup> Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; eman.kms@tu.edu.sa

<sup>3</sup> Department of Computer Sciences, College of Computer and Information Science, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia; mfhaojouni@pnu.edu.sa

\* Correspondence: zaali@ssuet.edu.pk

**Abstract:** This research offers an improved method for the self-organization of a swarm of UAVs based on a social learning approach. To start, we use three different colonies and three best members i.e., unmanned aerial vehicles (UAVs) randomly placed in the colonies. This study uses max-min ant colony optimization (MMACO) in conjunction with social learning mechanism to plan the optimized path for an individual colony. Hereinafter, the multi-agent system (MAS) chooses the most optimal UAV as the leader of each colony and the remaining UAVs as agents, which helps to organize the randomly positioned UAVs into three different formations. Afterward, the algorithm synchronizes and connects the three colonies into a swarm and controls it using dynamic leader selection. The major contribution of this study is to hybridize two different approaches to produce a more optimized, efficient, and effective strategy. The results verify that the proposed algorithm completes the given objectives. This study also compares the designed method with the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) to prove that our method offers better convergence and reaches the target using a shorter route than NSGA-II.

**Keywords:** social learning; ant colony optimization; multi-agent system



**Citation:** Shafiq, M.; Ali, Z.A.; Israr, A.; Alkhamash, E.H.; Hadjouni, M. A Multi-Colony Social Learning Approach for the Self-Organization of a Swarm of UAVs. *Drones* **2022**, *6*, 104. <https://doi.org/10.3390/drones6050104>

Academic Editor: Xiwang Dong

Received: 11 April 2022

Accepted: 20 April 2022

Published: 23 April 2022

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



**Copyright:** © 2022 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 has been exponentially increasing in the domains of flight control, path planning, and obstacle avoidance of unmanned aerial vehicles (UAVs) [1–3]. The analyses get increasingly complex when dealing with multiple UAVs in different formations. The natural behaviors of birds, ants, and fishes have been proven to be significant in formulating successful bio-inspired algorithms for the formation control, route planning, and trajectory tracking of a swarm of multiple UAVs [4–6]. Some of the important algorithms inspired by nature include ant colony optimization [7], pigeon-inspired optimization [8], and particle swarm optimization [9].

The primary inspiration for this study is to utilize the knowledge obtained from studying the natural flocking and swarming activities of ants and use them for controlling UAVs. Researchers have used these nature-inspired algorithms for numerous purposes including cooperative path planning of multiple UAVs [10], distributed UAV flocking among obstacles [11], and forest fire fighting missions [12]. We also find multiple studies that hybridize a bio-inspired algorithm with another method to increase its efficiency [13–15].

There are many existing solutions regarding the problems of path planning and multi-UAV cooperation. One such research study [16] deals with the inspection of an oilfield using multiple UAVs while avoiding obstacles. The researchers achieve this using an improved version of Non-Dominated Sorting Genetic Algorithm (NSGA). Another existing solution to tackle a multi-objective optimization is addressed in reference [17]. In [17],

the researchers use a hybrid of NSGA and local fruit fly optimization to solve interval multi-objective optimization problems. In reference [18], academics use a modified particle swarm optimization algorithm for the dynamic target tracking of multiple UAVs.

Our proposed method consists of many concepts, which are explained as follows:

Ant colony optimization (ACO) is an optimization technique used by ant colonies to find the shortest route to take to get to their food [19]. Using pheromones left behind from earlier ants, the ACO mimics ants looking for food. The path used by the most ants contains the most pheromones, which aids the next ant in choosing the shortest way [20]. The ACO technique is used in a variety of applications, such as the routing of autonomous vehicles and robots, which need to find the shortest path to a destination, and the design of computer algorithms, which need to find the optimal solution for a given problem.

Sometimes, however, the ACO is slow to converge and falls into the local optimum. To help solve these issues, researchers introduced a modified version of ACO called max-min ant colony optimization (MMACO). It operates by controlling the maximum and minimum amounts of pheromone that can be left on each possible trial [21]. The range of possible pheromone amounts on each possible route is limited to avoid stagnation in the search process.

In social animals, social learning plays a significant part in behavior learning. Social learning, as opposed to asocial (individual) learning, allows individuals to learn from the actions of others without experiencing the costs of individual trials and errors [22]. That is why this study incorporates social learning mechanisms into MMACO. Unlike traditional MMACO variations, which update ants based on past data, each ant in the proposed SL-MMACO learns from any better ants (called demonstrators) in the present swarm.

Some of the state-of-the-art work in the field of optimization algorithms include research [23] that proposes the use of social learning-based particle swarm optimization (SL-PSO) for integrated circuits manufacturing. The SL-PSO is used to increase the imaging performance in extreme ultraviolet lithography. Results showed that the errors were reduced significantly compared to conventional methods. Similarly, another recent study [24] uses improved ant colony optimization (IACO) for human gait recognition. The IACO is used to enhance the extracted features, which are then passed on to the classifier. Compared with current methods, the IACO technique in [24] is more accurate and takes less time to compute.

A multi-agent system (MAS) is a collection of agents that interact with each other and the environment to achieve a common goal. The main function of the MAS is to tackle issues that a single agent would find difficult to solve. To achieve its objective, another important function of the MAS is to be able to interact with each agent and respond accordingly. In MAS, each agent can determine its state and behavior based on the state and behavior of its neighbors [25]. MAS has multiple uses in fields such as robotics, computer vision, and transportation [26–28]. In most MAS scenarios, an external entity is required to direct the agents toward the destination [29].

A leader is an agent that can alter the states of the follower agents. A leader can be outside or inside the MAS or can even be virtual. A large swarm of UAVs can be controlled more efficiently by selecting fewer leaders than follower agents. For example, when we want to control a fleet of UAVs, we can select a few leader agents, let the remaining UAVs follow the leader, and use the leading UAVs to control the state of the system.

The major contributions of this study are as follows:

1. Designing a novel hybrid algorithm that combines MMACO with the social learning mechanism to enhance its performance.
2. Using the designed algorithm in tandem with the MAS and dynamic leader selection to connect the three colonies.
3. Achieving the self-organization of the three colonies and synchronizing them into one swarm.
4. Demonstrating the validity of the designed method by performing computer simulations that mimic real-life scenarios.

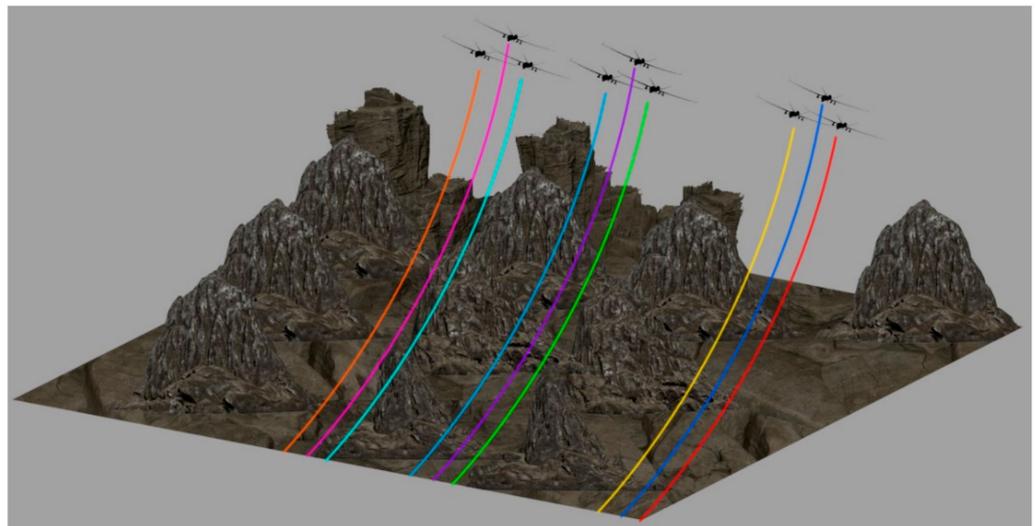
The paper is organized into seven sections. Section 1 presents the introduction and the literature review. In Section 2, we break down the problem into three scenarios and describe each one in detail. Section 3 provides the framework of the proposed solution. Section 4 defines the proposed method with each of its constituent parts discussed at length and offers the algorithm and the flowchart. In Section 5, we discuss the simulations and their outcomes. Section 6 presents the conclusion of the study.

## 2. Research Design

We broke our problem into three different scenarios to help make it easier. In the first scenario, the UAVs are in random positions and then using our proposed algorithm, they organize themselves into a formation. In the second scenario, we navigate the newly organized formations through some obstacles. In the last scenario, we combine the three formations into one swarm and then navigate it through the same environment. Below, we describe each scenario in detail:

### Scenario 1:

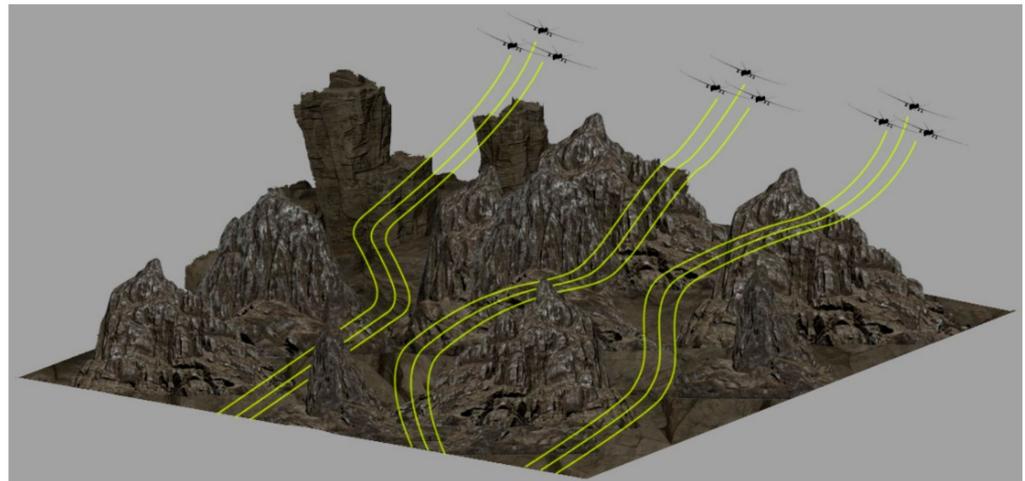
Figure 1 presents the first scenario. In this scenario, there are three UAVs in three different colonies and the UAVs are placed at random positions within each colony. The environment contains different obstacles like mountains and rough terrain. The goal is to reach the target using the shortest route possible without colliding with other UAVs. The main objective here is to maintain the formation throughout the journey.



**Figure 1.** Illustration of the first scenario.

### Scenario 2:

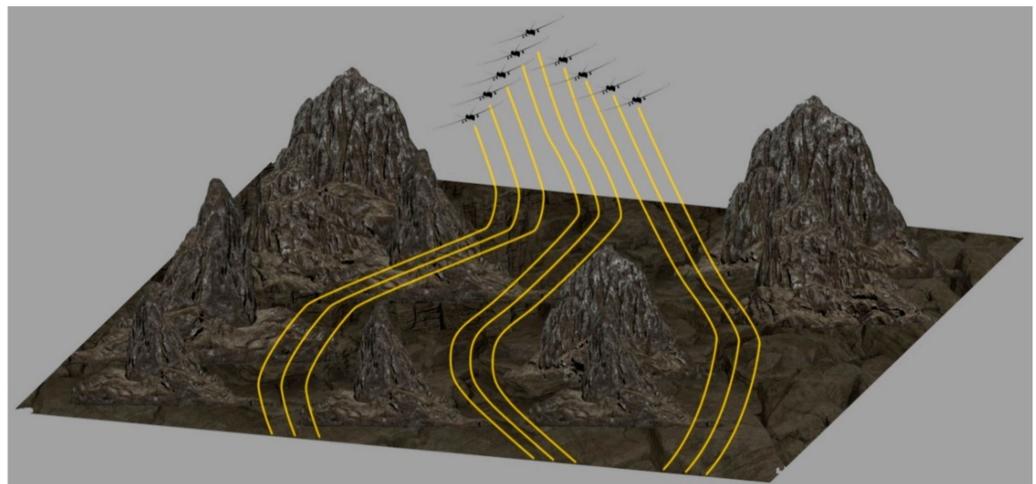
Figure 2 illustrates the second scenario. In this scenario, there are again three UAVs in three different colonies and the UAVs are placed at random positions within each colony. The environment is also the same. The goal is to reach the target using the shortest route possible without colliding with the obstacles or other UAVs. The main distinction between the first and the second scenario is that, in the first task, we only had to demonstrate the ability of the algorithm to maintain a formation. However, the second task also requires navigating through the obstacles without any collision.



**Figure 2.** Illustration of the second scenario.

### Scenario 3:

Figure 3 illustrates the third scenario. In this scenario, we pick up where the second scenario left off, i.e., the three colonies are now in the desired formations. The environment is the same as in the second scenario. The goal is to first synchronize the three colonies into one big swarm, and then, while maintaining the swarm, reach the target using the shortest route possible without colliding with the obstacles or other UAVs.



**Figure 3.** Illustration of the third scenario.

### 3. Solution Architecture

Figure 4 presents the framework of our proposed solution for the aforementioned problems. It is clear from the figure that, initially, the UAVs in the three different colonies are at random positions. Here, we apply the social learning-based max-min ant colony optimization (SL-MMACO) to each colony. SL-MMACO works by first finding the most optimal routes for each colony, and then the social learning mechanism sorts the ants from best to worst. Afterward, the multi-agent system (MAS) appoints the best ant as the leader and the remaining ants as agents. In the next step, we see that all the colonies are now synchronized to avoid any collision between the UAVs. Finally, we connect all three colonies into one big swarm and then select its leader dynamically according to the mission requirements.

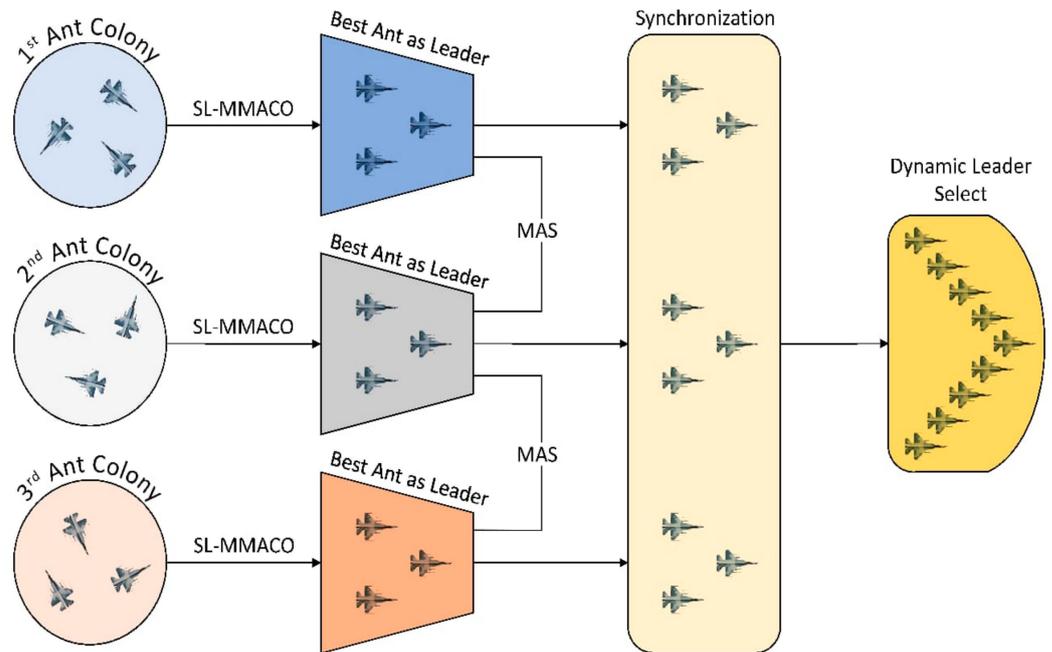


Figure 4. Solution architecture.

#### 4. Proposed Algorithm

This section introduces the different concepts and theories that we used for the development of our proposed algorithm. In this research paper, we are using a graph-based approach.

##### 4.1. Ant Colony Optimization

Here, we are using a graph-based approach. The concept of nodes, edges, and legs is important to understand path planning using ant colony optimization (ACO). Figure 5 presents the relationship between the nodes, edges, and legs. The ACO generates intermediary points between the initial and final positions. These intermediate points are called the nodes. An edge is a link between two nodes. For instance, edge (b,c) is the length from edge b to c, whereas a leg is generated whenever a UAV turns.

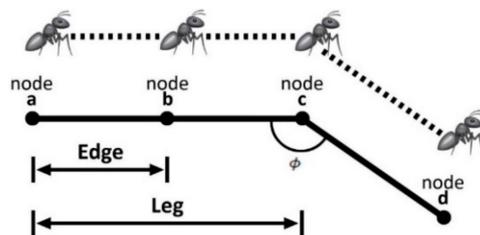


Figure 5. Relationship between nodes, edges, and legs.

Suppose that the  $m$ th ant is at node  $i$  on time  $t$ , the probability of transition can be written as:

$$p_{ij}^m(t) = \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{c \in allowed_i} \tau_{ic}^\alpha \eta_{ic}^\beta} \quad (1)$$

where the probability of transition from node  $i$  to node  $j$  of the  $m$ th ant is  $p_{ij}^m(t)$ , the pheromone on the edge  $(i, j)$  is  $\tau_{ij}(t)$ , the transit feasibility from node  $i$  to node  $j$  is  $\eta_{ij}(t)$ , the set of nodes that are neighboring  $i$  is  $allowed_i$ , the constant influencing the  $\tau_{ij}(t)$  is  $\alpha$ , and the constant influencing the  $\eta_{ij}(t)$  is  $\beta$ .

After the method begins, the starting pheromone rate varies according to the edges. The pheromone rate is then reset by each ant that generated the result, which starts the next cycle of the process.  $\tau_{ij}(t)$  on the edge  $(i, j)$  is:

$$\tau_{ij}(t + 1) = (1 - \rho) \times \tau_{ij}(t) + \sum_{m=1}^k \Delta\tau_{ij}^m(t) \tag{2}$$

where the rate of pheromone evaporation is  $\rho$  ( $0 \leq \rho \leq 1$ ), total ants are represented by  $k$ , and the pheromone rate of the edge  $(i, j)$  is  $\Delta\tau_{ij}^m(t)$ .  $\Delta\tau_{ij}^m(t)$  can be further defined as

$$\Delta\tau_{ij}^m(t) = \begin{cases} Q/L_m; & \text{ant 'm' uses edges of (i, j)} \\ 0; & \text{otherwise} \end{cases} \tag{3}$$

where the length of the route built by the  $m$ th ant is  $L_m$  and  $Q$  is the constant.

#### 4.2. Max-Min Ant Colony Optimization

We need to improve the traditional method to ensure that the ACO converges quickly. MMACO delivers some remarkable results in this area by restricting the pheromones on each route. To understand MMACO, we must first examine the path's cost. The average cost of path  $J_{a,k}(t)$  can be given as:

$$J_{a,k}(t) = \frac{1}{k} \sum_{m=1}^k J_{a,m}(t) \tag{4}$$

Note that the  $m$ th ant only updates the pheromone when the cost of path of the  $m$ th ant in the  $t$ th iteration fulfill  $J_{a,k}(t) \geq J_{a,m}(t)$ .

The MMACO updates the route using Equation (3) after every iteration. After each iteration, the algorithm determines the most optimum and least optimal paths. To improve the probability of discovering the global best route, it discards the least optimum route. As a result, Equation (3) may be revised as follows:

$$\Delta\tau_{ij}^m(t) = \begin{cases} Q/L_o; & \text{route (i, j) refer to the optimal route} \\ -Q/L_w; & \text{route (i, j) refer to the worst route} \\ 0; & \text{otherwise} \end{cases} \tag{5}$$

In the above equation,  $L_o$  is the most optimal route and  $L_w$  is the current iteration's worst route. The quantity of pheromone produced by MMACO is limited to specified values. This limitation aids in accelerating convergence and avoiding stagnation.

The algorithm restricts the pheromone on each route to a specified minimum and maximum value, denoted by  $\tau_{min}$  and  $\tau_{max}$ , respectively. This can be represented mathematically as:

$$\tau_{ij}(t) = \begin{cases} \tau_{max}; & \tau_{ij}(t) \geq \tau_{max} \\ \tau_{ij}(t); & \tau_{min} < \tau_{ij}(t) < \tau_{max} \\ \tau_{min}; & \tau_{ij}(t) \leq \tau_{min}(t) \end{cases} \tag{6}$$

#### 4.3. Social Learning Mechanism

The conduct of a person to learn from their surroundings is referred to as social learning. You should learn not just from the top students in class, but also from students who are better than you. Most biological groups follow the same idea. Initially, for the map and compass process, locations and velocities of the ants are produced at random and are indicated as  $X_i$  and  $V_i$  ( $i = 1, 2, \dots, m$ ). At the next iteration, the new location  $X_i$  and velocity  $V_i$  are calculated by the formula [30]:

$$X_a^{N_c} = X_a^{N_c-1} + V_a^{N_c} \tag{7}$$

$$V_a^{N_c} = V_a^{N_c-1} \times e^{-R.N_c} + rand \times c_1 \times (X_{mod} - X_a^{N_c-1}) \tag{8}$$

$$c_1 = 1 - \log\left(\frac{N_c}{m}\right) \tag{9}$$

where  $R$  represents the map and compass factor, which is between 0 and 1,  $N_c$  is the current number of iterations,  $c_1$  the learning factor,  $X_{mod}$  the demonstrator ant superior to the current ant, and  $m$  is the total number of ants. Each follows the ant better than itself, and this is known as the learning behavior. Figure 6 illustrates the process for the selection of demonstrator  $X_{mod}$ .

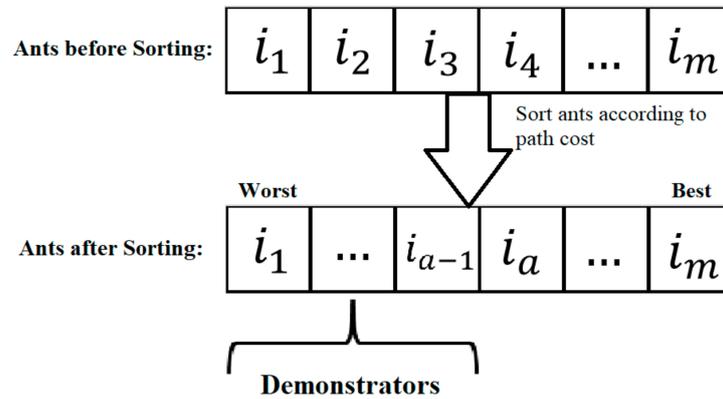


Figure 6. Social learning mechanism.

For the landmark operation, the social behavior occurs when ants from the center are removed, and other ants migrate toward the center. The procedure can be given as:

$$X_{cent}^{N_c-1} = \sum_{a=1}^{N_c-1} \frac{X_a^{N_c-1}}{N^{N_c-1}} \tag{10}$$

$$X_a^{N_c} = X_a^{N_c-1} + rand \times c_2 \times (X_{cent}^{N_c-1} - X_a^{N_c-1}) \tag{11}$$

$$c_2 = \alpha_s \left(\frac{N_c}{m}\right) \tag{12}$$

where the social influence factor is  $c_2$ , and  $\alpha_s$  is called the social coefficient.

#### 4.4. Multiple Agent Systems

The multi-agent system comprises  $n$  independent agents that move with the same absolute velocity. Every agent’s direction is updated to its neighbor’s status. At time  $t$ , the neighbors of an agent  $A$  ( $1 \leq A \leq n$ ) are those who are located within a circle of radius  $r$  ( $r > 0$ ) centered on the position of agent  $a$ . At time  $t$ , the neighbor of the agent  $a$  is  $N_a(t)$ ,

$$N_a(t) = \{b | d_{ab}(t) < r\} \tag{13}$$

where the Pythagorean Theorem can be used to compute  $d_{ab}(t)$ .

The coordinates of the agent at time  $t$  are  $(x_a(t), y_a(t))$ , where agent  $a$  is a neighbor to agent  $b$ . The absolute velocity  $v$  ( $v > 0$ ) of each agent in the system is the same.

$$x_a(t + 1) = x_a(t) + v \cos \theta_a(t) \tag{14}$$

$$y_a(t + 1) = y_a(t) + v \sin \theta_a(t) \tag{15}$$

At time  $t$ , the heading angle of agent  $a$  is  $\theta_a(t)$ . The following equation is used by the algorithm to update the heading angle:

$$\theta_a(t + 1) = \tan^{-1} \frac{\sum_{b \in N_a(t)} \sin \theta_b(t)}{\sum_{b \in N_a(t)} \cos \theta_b(t)} \tag{16}$$

The equation above is used to discover obstructions by examining their surroundings. If a missing node in the neighbor (i.e., a hurdle) exists, the heading angle will be changed to prevent colliding with the obstruction.

We can analyze this algorithm using basic graph theory. Please note that each agent’s neighbors are not always the same. The undirected graph set  $\mathbb{G}_t = \{\mathcal{V}, \varepsilon_t\}$  is used for agent coordination. Where the set containing every agent is  $\mathcal{V} = \{1, 2, \dots, N\}$ , and the time-varying edge set is  $\varepsilon_t$ . A graph is connected if any two of its vertices are connected.

Equation (16) can be modified as:

$$\tan \theta_a(t + 1) = \sum_{b \in N_a(t)} \frac{\cos \theta_b(t)}{\sum_{m \in N_a(t)} \cos \theta_m(t)} \tan \theta_a(t) \tag{17}$$

To further simplify Equation (17), we use a matrix,

$$\tan \theta(t + 1) = I(t) \tan \theta(t) \tag{18}$$

where  $\tan \theta(t) \triangleq (\tan \theta_1(t), \dots, \tan \theta_N(t))^T$ . For the graph  $\mathbb{G}_t$ , the weighted average matrix is  $I(t) \triangleq (i_{ab}(t))$ .

$$i_{ab}(t) = \begin{cases} \frac{\cos \theta_b(t)}{\sum_{m \in N_a(t)} \cos \theta_m(t)} & \text{if } (a, b) \in \varepsilon_t \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

For the synchronization, we study the linear model of Equation (16) as follows:

$$\theta_a(t + 1) = \frac{1}{n_a(t)} \sum_{b \in N_a(t)} \theta_b(t) \tag{20}$$

where the number of elements in  $N_a(t)$  is  $n_a(t)$ . Equation (18) can be rewritten as,

$$\tan \theta(t + 1) = \tilde{I}(t) \theta(t) \tag{21}$$

where  $\theta(t) \triangleq (\theta_1(t), \dots, \theta_N(t))^T$ , and the entries of the matrix  $\tilde{I}(t)$  are,

$$\tilde{i}_{ab}(t) = \begin{cases} \frac{1}{n_a(t)}, & \text{if } (a, b) \in \varepsilon_t \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

#### 4.5. Synchronization and Connectivity

To continue the study of the synchronization of the designed algorithm and the connectivity of the associated neighboring graphs, we should formally describe synchronization. If the headings of each agent match the following criteria, the system will achieve synchronization.

$$\lim_{t \rightarrow \infty} \theta_a(t) = \theta, a = 1, \dots, N \tag{23}$$

where  $\theta$  varies to the starting values  $\{\theta_a(0), x_a(0), y_a(0), a = 1, \dots, N\}$  and the system parameters  $v$ , and  $r$ .

Considering the model in Equations (14)–(16), let  $\{\theta_a(0) \in (-\frac{\pi}{2}, \frac{\pi}{2}), a = 1, \dots, N\}$ , and assume that the neighbor at the start,  $\mathbb{G}_0 = \{\mathcal{V}, \varepsilon_0\}$  is connected. Therefore, to achieve synchronization, the system will have to satisfy,

$$v \leq \frac{d}{\Delta_0} \left( \frac{\cos \bar{\theta}}{N} \right)^N \tag{24}$$

whereas the number of agents is represented by  $N$ . Meanwhile,

$$\bar{\theta} = \max_a |\theta_a(0)| \tag{25}$$

$$d = r - \max_{a,b \in \varepsilon_0} d_{ab}(0) \tag{26}$$

$$\Delta_0 = \max_{a,b} \{ \tan \theta_a(0) - \tan \theta_b(0) \} \tag{27}$$

Considering the models in Equations (14), (15) and (20), let  $\theta_a(0) \in [0, 2\pi)$ , and let us assume that the starting neighbor graph is connected. Therefore, to achieve synchronization, the system will have to satisfy,

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

whereas  $d$  is the same as described in Equation (26).

#### 4.6. Dynamic Leader Selection

Due to communication problems between agents, the formation structure of multi-agent systems might occasionally change. Considering a random failure of communication, each connection  $(a,b) \in E$  fails independently with probability  $p$ . Let  $\hat{G}$  be the graph topology and  $E_{\hat{G}}(t_{conv})$  be the expected convergence time for this model of communication failure, while  $E_{\hat{G}}(\cdot)$  denotes the argument's expectation across the group of network structures represented by  $\hat{G}$ . The convergence rate will be maximized by reducing  $E_{\hat{G}}(t_{conv})$ . As a result, the formula for choosing a leader  $k$  to maximize the convergence rate is as follows:

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

So that it meets the below conditions,

$$\begin{cases} \text{tr}(Y) \geq n - k \\ Y_{aa} \in \{0, 1\} \forall a \neq V \\ Y_{ab} = 0 \forall a \neq b \end{cases} \tag{30}$$

As such, the objective function is in line with the projected convergence rate for potential network topologies. Where  $\min_{x(0)} x(0)^T (Y\hat{L} + \hat{L}Y)x(0)$  is a convex function of  $Y$ .

#### 4.7. B-Spline Path Smoothing

The hybrid algorithm's route consists mostly of a combination of line segments. The B-spline curve method is utilized to ensure the smoothness of the route created. The B-spline method is an improvement over the Bezier approach that preserves the convexity and geometrical invariability.

The B-spline path smoothing can be written as:

$$P(u) = \sum_{i=0}^n d_i N_{ij}(u) \tag{31}$$

Considering Equation (31),  $d_i (i = 0, 1, \dots, n)$  are control points, and  $N_{ij}(u)$  are the normalized b-order functions of the B-spline. These can be described as:

$$\begin{cases} N_{ij}(u) = \begin{cases} 1, & \text{if } u_i \leq u \leq u_{i+1} \\ 0, & \text{otherwise} \end{cases} \\ N_{ij}(u) = \frac{u-u_i}{u_{i+j}-u_i} N_{i,j-1}(u) + \frac{u_{i+j+1}-u}{u_{i+j+1}-u_{i+1}} N_{i+1,j+1}(u) \\ \text{define } N_{i0} = 0 \end{cases} \tag{32}$$

The essential functions of the B-spline curve are determined by the parametric knots  $\{u_0 \leq u_1 \leq \dots \leq u_{n+j}\}$ . In contrast to the Bezier curve, the B-spline curve is unaffected by altering a single control point. Another benefit of the control points over the Bezier curve is that the degree of polynomials does not increase when the control points are increased.

#### 4.8. Flowchart and Algorithm

Figure 7 illustrates the flowchart of the whole system. As we can see, the system first initializes the parameters in all colonies and places ants at the starting random positions. Then, the ants keep moving to the next node until they reach the destination. Hereinafter, they compute the path cost and update the pheromone. However, the system only updates the path if the pheromone is within the desired range. This process is repeated for the predefined number of iterations. Next, the social learning mechanism sorts the ants from best to worst according to the path cost, and the algorithm assigns the best ant to be the leader of each colony and the remaining ants to be agents. Finally, all the colonies are synchronized into one big swarm, and the system dynamically selects its leader.

---

**Algorithm 1** Designed Strategy is Given As.

---

```

Parameter Initialization of the algorithm
procedure algorithm1
  while ( $N_c < N_{c_{max}}$ ) do
    Put virtual ants on initial positions
    If (Destination_Arrived) then
      Compute the average cost of the path using Equation (4) and update the phenomenon
      using Equations (2) and (3) and (5) and (6)
    else
      Move to the next node using Equation (1)
    end if
    Sort the ants according to the Social Learning mechanism using Equations (7)–(12)
    Output the best ant
  end while
  while (destination) do
    if (Found_Obstacle) then
      Update heading angle using Equation (16)
    else
      Keep moving towards the destination
    end if
    while (Total_UAVs)
      Dynamically select the leader using Equations (29) and (30)
    end while
  end while
end procedure

```

---

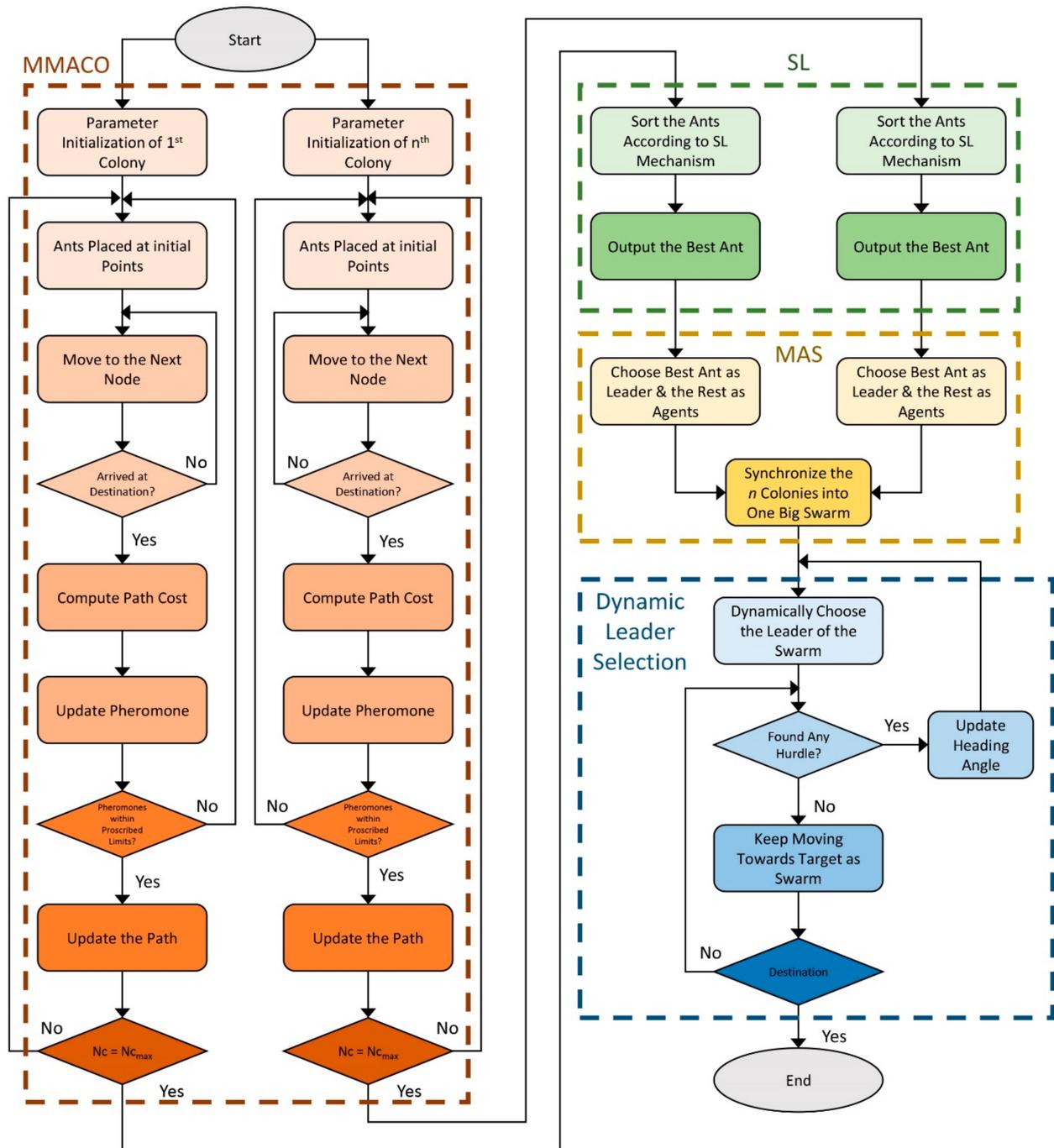


Figure 7. Illustration of the flowchart.

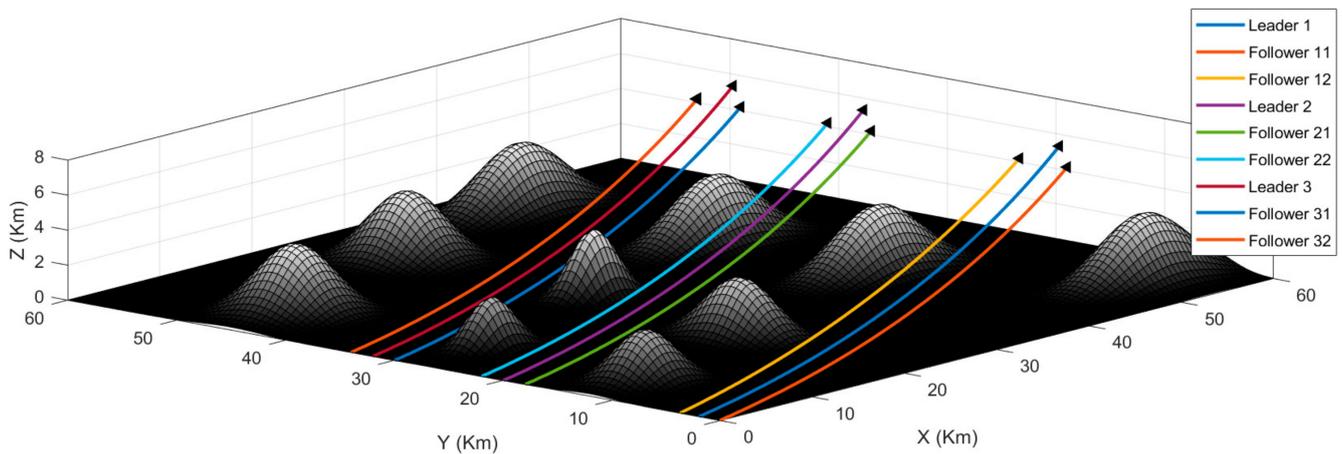
### 5. Simulation Results

In this section, we verify the effectiveness of the proposed strategy by applying it to a MATLAB simulation. The conclusions that we want to verify are the following: firstly, we apply the algorithm to three colonies with each having three UAVs, and see if they can organize themselves into desired formations. For the second scenario, we navigate these newly organized formations through some obstacles to see if they can maintain their formations. Lastly, we validate if the designed strategy can successfully synchronize and connect the three colonies into one swarm.

First Scenario:

For the first scenario, the UAVs are at random positions within each colony. The objective is to arrange the UAVs into the desired formations and then reach the target

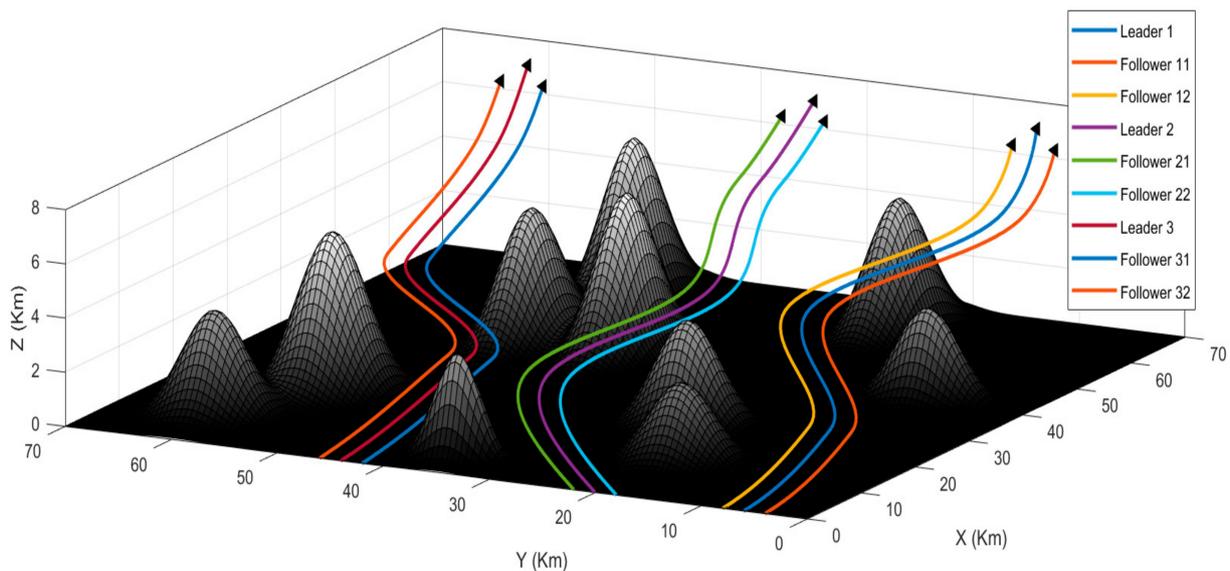
using the shortest possible path. Figure 8 presents the simulation result of the first scenario. As we can see, the algorithm successfully arranges the UAVs into formations, and they maintain these formations throughout the journey. The environment contains different obstacles like mountains and rough terrain. Please note that in this scenario, we are only testing the capability of the algorithm to maintain formations, and hence, we fly the UAV formations in an upward trajectory and not through the obstacles. The second and the third scenarios deal with passing the formations through the obstacles.



**Figure 8.** Simulation result of the first scenario.

#### Second Scenario:

In this scenario, after the algorithm successfully maintains the formation, and we check whether it is also capable of obstacle avoidance. The environment contains different obstacles like mountains and rough terrain. The goal is to reach the target using the shortest route possible without colliding with the obstacles or other UAVs. Figure 9 illustrates the simulation result of the second scenario. It is evident from the result that the algorithm achieved the desired goal, and we can see that the three colonies navigated the obstacles while maintaining formations.

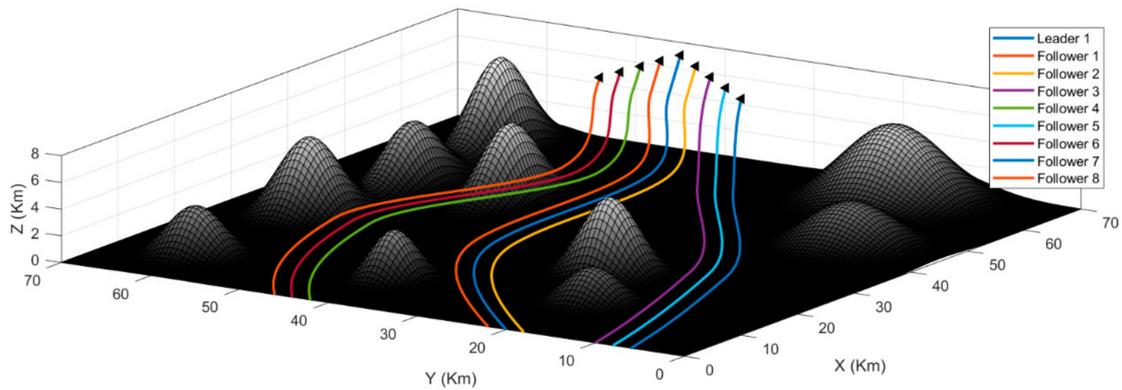


**Figure 9.** Simulation result of the second scenario.

#### Third Scenario:

In this scenario, we pick up where the second scenario left off, i.e., the three colonies are now in the desired formations. The environment contains different obstacles like mountains

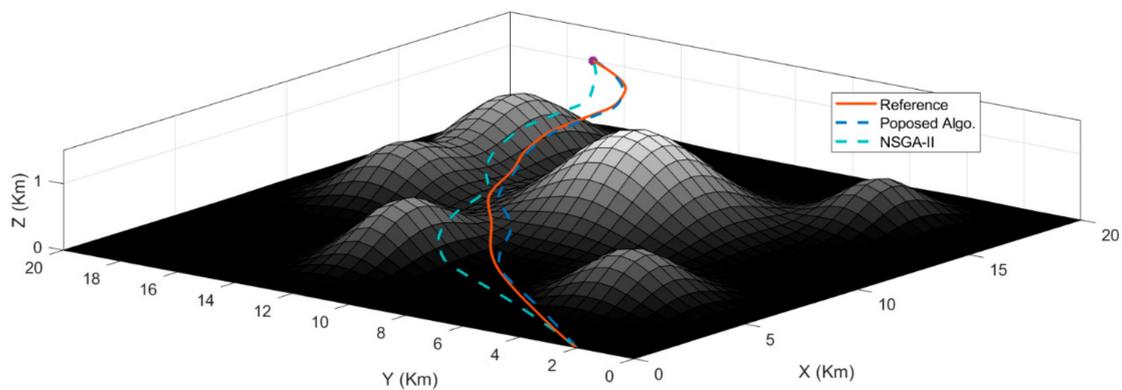
and rough terrain. The goal is to first synchronize the three colonies into one big swarm, and then while maintaining the swarm, reach the target using the shortest route possible without colliding with the obstacles or other UAVs. Figure 10 illustrates the simulation result of the third scenario. Again, we see that the algorithm successfully synchronized the three colonies into one swarm, and it reaches its target without any collision.



**Figure 10.** Simulation result of the third scenario.

#### Comparison with NSGA-II:

Lastly, we also compare our proposed algorithm with the Non-Dominated Sorting Genetic Algorithm II (NSGA II). We compare our proposed method with the NSGA II because it is a fast multi-objective genetic algorithm and is a highly regarded evolutionary algorithm. Figure 11 compares our proposed method with NSGA-II. Here, we can see that our designed strategy stays close to the reference while NSGA-II sometimes strays too far. Additionally, our strategy follows a shorter and quicker path to the target.



**Figure 11.** Comparison of our proposed method with NSGA-II.

## 6. Conclusions

This research presents a strategy for the self-organization of a swarm of UAVs consisting of three colonies with three UAVs each. To plan the path of each colony to the target, we used max-min ant colony optimization (MMACO), and we used the social learning mechanism to sort the ants from worst to best. To organize the randomly positioned UAVs into different formations, this study used the multi-agent system (MAS). The designed algorithm also synchronized and connected the three colonies into a swarm with the help of dynamic leader selection. The proposed algorithm completed the given objectives in the simulation results.

The salient results and the findings in this research include successfully maintaining the formations in the first scenario. In the second scenario, the algorithm not only maintained the formations, but also navigated them through the obstacles. In the third scenario,

the algorithm merged the three formations into one big swarm and then successfully navigated the swarm through the obstacles. By comparing the proposed method with the Non-Dominated Sorting Genetic Algorithm II (NSGA-II), it was clear that our strategy offered better convergence, optimized routes, and reached the destination using a shorter route than NSGA-II.

**Author Contributions:** Investigation, M.S.; supervision, Z.A.A.; validation, A.I.; methodology, E.H.A.; writing-original draft preparation, M.S.; software, Z.A.A.; funding acquisition, E.H.A.; writing-review and editing, A.I. and M.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by Taif University Researchers Supporting Project number (TURSP-2020/292) Taif University, Taif, Saudi Arabia and this work is also supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R193), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

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

**Acknowledgments:** The authors would like to acknowledge Taif University Researchers Supporting Project number (TURSP-2020/292) Taif University, Taif, Saudi Arabia. The authors would like also to acknowledge, Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R193), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

**Conflicts of Interest:** The authors declare that they have no conflict of interest.

## References

1. Radmanesh, M.; Kumar, M.; Guentert, P.H.; Sarim, M. Overview of path-planning and obstacle avoidance algorithms for UAVs: A comparative study. *Unmanned Syst.* **2018**, *6*, 95–118. [[CrossRef](#)]
2. Yao, P.; Wang, H.; Su, Z. Real-time path planning of unmanned aerial vehicle for target tracking and obstacle avoidance in complex dynamic environment. *Aerosp. Sci. Technol.* **2015**, *47*, 269–279. [[CrossRef](#)]
3. Seo, J.; Kim, Y.; Kim, S.; Tsourdos, A. Collision avoidance strategies for unmanned aerial vehicles in formation flight. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 2718–2734. [[CrossRef](#)]
4. Oh, H.; Shirazi, A.R.; Sun, C.; Jin, Y. Bio-inspired self-organising multi-robot pattern formation: A review. *Robot. Auton. Syst.* **2017**, *91*, 83–100. [[CrossRef](#)]
5. Ganesan, R.; Raajini, X.M.; Nayyar, A.; Sanjeevikumar, P.; Hossain, E.; Ertas, A.H. Bold: Bio-inspired optimized leader election for multiple drones. *Sensors* **2020**, *20*, 3134. [[CrossRef](#)] [[PubMed](#)]
6. Xie, Y.; Han, L.; Dong, X.; Li, Q.; Ren, Z. Bio-inspired adaptive formation tracking control for swarm systems with application to UAV swarm systems. *Neurocomputing* **2021**, *453*, 272–285. [[CrossRef](#)]
7. Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. [[CrossRef](#)]
8. Duan, H.; Qiao, P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. *Int. J. Intell. Comput. Cybern.* **2014**, *7*, 24–37. [[CrossRef](#)]
9. Poli, R.; Kennedy, J.; Blackwell, T. Particle swarm optimization. *Swarm Intell.* **2007**, *1*, 33–57. [[CrossRef](#)]
10. Ali, Z.A.; Han, Z.; Wang, B.H. Cooperative path planning of multiple UAVs by using max–min ant colony optimization along with Cauchy mutant operator. *Fluct. Noise Lett.* **2021**, *20*, 2150002. [[CrossRef](#)]
11. Qiu, H.; Duan, H. A multi-objective pigeon-inspired optimization approach to UAV distributed flocking among obstacles. *Inf. Sci.* **2020**, *509*, 515–529. [[CrossRef](#)]
12. Ghamry, K.A.; Kamel, M.A.; Zhang, Y. Multiple UAVs in forest fire fighting mission using particle swarm optimization. In Proceedings of the 2017 IEEE International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 1404–1409.
13. Vijh, S.; Gaurav, P.; Pandey, H.M. Hybrid bio-inspired algorithm and convolutional neural network for automatic lung tumor detection. *Neural Comput. Appl.* **2020**, 1–14. [[CrossRef](#)]
14. Domanal, S.G.; Guddeti, R.M.R.; Buyya, R. A hybrid bio-inspired algorithm for scheduling and resource management in cloud environment. *IEEE Trans. Serv. Comput.* **2017**, *13*, 3–15. [[CrossRef](#)]
15. Dhiman, G. ESA: A hybrid bio-inspired metaheuristic optimization approach for engineering problems. *Eng. Comput.* **2021**, *37*, 323–353. [[CrossRef](#)]
16. Li, K.; Yan, X.; Han, Y.; Ge, F.; Jiang, Y. Many-objective optimization based path planning of multiple UAVs in oilfield inspection. *Appl. Intell.* **2022**, 1–16. [[CrossRef](#)]
17. Ge, F.; Li, K.; Han, Y. Solving interval many-objective optimization problems by combination of NSGA-III and a local fruit fly optimization algorithm. *Appl. Soft Comput.* **2022**, *114*, 108096. [[CrossRef](#)]

18. Wang, Y.; Li, K.; Han, Y.; Yan, X. Distributed multi-UAV cooperation for dynamic target tracking optimized by an SAQPSO algorithm. *ISA Trans.* 2021, in press. [\[CrossRef\]](#)
19. Shetty, A.; Shetty, A.; Puthusseri, K.S.; Shankaramani, R. An improved ant colony optimization algorithm: Minion Ant (MAnt) and its application on TSP. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; pp. 1219–1225.
20. Shi, B.; Zhang, Y. A novel algorithm to optimize the energy consumption using IoT and based on Ant Colony Algorithm. *Energies* **2021**, *14*, 1709. [\[CrossRef\]](#)
21. Shafiq, M.; Ali, Z.A.; Alkhamash, E.H. A cluster-based hierarchical-approach for the path planning of swarm. *Appl. Sci.* **2021**, *11*, 6864. [\[CrossRef\]](#)
22. Cheng, R.; Jin, Y. A social learning particle swarm optimization algorithm for scalable optimization. *Inf. Sci.* **2015**, *291*, 43–60. [\[CrossRef\]](#)
23. Zhang, Z.; Li, S.; Wang, X.; Cheng, W.; Qi, Y. Source mask optimization for extreme-ultraviolet lithography based on thick mask model and social learning particle swarm optimization algorithm. *Opt. Express* **2021**, *29*, 5448–5465. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Khan, A.; Javed, M.Y.; Alhaisoni, M.; Tariq, U.; Kadry, S.; Choi, J.; Nam, Y. Human Gait Recognition Using Deep Learning and Improved Ant Colony Optimization. *Comput. Mater. Cont.* **2022**, *70*, 2113–2130. [\[CrossRef\]](#)
25. Kamdar, R.; Paliwal, P.; Kumar, Y. A state of art review on various aspects of multi-agent system. *J. Circuits Syst. Comput.* **2018**, *27*, 1830006. [\[CrossRef\]](#)
26. Gulzar, M.M.; Rizvi, S.T.H.; Javed, M.Y.; Munir, U.; Asif, H. Multi-agent cooperative control consensus: A comparative review. *Electronics* **2018**, *7*, 22. [\[CrossRef\]](#)
27. González-Briones, A.; Villarrubia, G.; De Paz, J.F.; Corchado, J.M. A multi-agent system for the classification of gender and age from images. *Comput. Vis. Image Underst.* **2018**, *172*, 98–106. [\[CrossRef\]](#)
28. Hamidi, H.; Kamankesh, A. An approach to intelligent traffic management system using a multi-agent system. *Int. J. Intell. Transp. Syst. Res.* **2018**, *16*, 112–124. [\[CrossRef\]](#)
29. Herrera, M.; Pérez-Hernández, M.; Parlikad, A.K.; Izquierdo, J. Multi-agent systems and complex networks: Review and applications in systems engineering. *Processes* **2020**, *8*, 312. [\[CrossRef\]](#)
30. Ruan, W.; Duan, H. Multi-UAV obstacle avoidance control via multi-objective social learning pigeon-inspired optimization. *Front. Inf. Technol. Electron. Eng.* **2020**, *21*, 740–748. [\[CrossRef\]](#)