applied sciences

MDPI

Article

A Cluster-Based Hierarchical-Approach for the Path Planning of Swarm

Muhammad Shafiq [1], Zain Anwar Ali [1,2,*] and Eman H. Alkhammash [3]

1 Electronic Engineering Department, Sir Syed University of Engineering & Technology, Karachi 75300, Pakistan; ssuet.shafiq@hotmail.com
2 School of Systems Science, Beijing Normal University, Zhuhai 519085, China
3 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
* Correspondence: zainanwar86@bnu.edu.cn

Abstract: This paper addresses the path planning and control of multiple colonies/clusters that have unmanned aerial vehicles (UAV) which make a network in a hazardous environment. To solve the aforementioned issue, we design a new and novel hybrid algorithm. As seen in the mission requirement, to combine the Maximum-Minimum ant colony optimization (ACO) with Vicsek based multi-agent system (MAS) to make an Artificially Intelligent (AI) scheme. In order to control and manage the different colonies, UAVs make a form of a network. The designed method overcomes the deficiencies of existing algorithms related to controlling and synchronizing the information globally. Furthermore, our designed architecture bounds, lemmatizes the pheromone, and finds the best ants which then make the most optimized path. The key contribution of this study is to merge two unique algorithms into a hybrid algorithm that has superior performance than both algorithms operating separately. Another contribution of the designed method is the ability to increase the number of individual agents inside the colony or the number of colonies with a good convergence rate. Lastly, we also compared the simulation results with the non-dominated sorting genetic algorithm II (NSGA-II) in order to prove the designed algorithm has a better convergence rate.

Keywords: bio-inspired algorithm; computer simulations; Max-Min ant colony optimization (MMACO); multi-agent systems (MAS)

Citation: Shafiq, M.; Ali, Z.A.; Alkhammash, E.H. A Cluster-Based Hierarchical-Approach for the Path Planning of Swarm. *Appl. Sci.* **2021**, *11*, 6864. https://doi.org/10.3390/app11156864

Academic Editor: José Manuel Andújar Márquez

Received: 1 June 2021
Accepted: 23 July 2021
Published: 26 July 2021

1. Introduction

Motivation: The motivation behind this study is to plan a path for a network of unmanned aerial vehicles (UAVs). It means designing a navigable path from the primary location to the destination it produces the most optimal outcome under specific constraints [1]. Many diverse research areas like engineering [2], science [3], wireless and cloud infrastructure [4–6], and even economics [7] encounter optimization problems. To resolve these optimization problems, researchers have offered numerous strategies [8–10]. As the problem size expands, these methods require more computations [11]. Hence, this paper aims to cover gaps in the research literature relating to optimization algorithms that not only require less memory and computational resources but also produce improved outcomes.

Related Works: A cluster is a set of similar objects and cluster analysis is used in a variety of different applications [12]. However, during the last few years, scientists combined different cluster-based strategies to provide more precise and robust results compared to their traditional counterparts [13–15]. Max-Min Ant Colony Optimization (MMACO) is one such optimization algorithm, which is an improvement over the classic Ant Colony Optimization (ACO) [16]. ACO a meta-heuristic algorithm that is used to solve complex optimization difficulties. As the name of the algorithm implies, a group of artificial ants searches for the best solutions through multiple iterations.

However, in some instances, ACO has a slow convergence rate and falls into the local optimum. To avoid the aforementioned issues, researchers introduced MMACO, a strategy that limits the pheromone on each trail to maximize the chances of getting an optimal result. MMACO differs from ACO in many significant aspects. For instance, MMACO only allows the best ants to add pheromone to the pheromone trail.

Multi-Agent System (MAS) contains multiple agents, and every agent determines its state using the state of its neighbor. MAS has numerous applications including in the fields of autonomous vehicles [17] and bio-inspired systems [18]. In most cases of MAS, an outside entity has to drive the agents toward the target. The usual strategy is to designate a few agents as leaders and let the rest of the agents follow the leader. The leader agents function as external entities that can affect the follower states. We can efficiently control a big network of UAVs by selecting fewer leader agents than the followers, which demonstrates the scalability and efficiency of the leader-follower method [19,20].

Vicsek et al. [21] introduced a comparable, simpler multi-agent model later known as the Vicsek model. It consists of discrete-time "N" autonomous agents having constant absolute velocity. These agents follow the direction of the motion of their neighbors also conjuction with random noises. Vicsek et al. [21] discovered some exciting results through simulations: all agents will follow the same direction when their density is large and the smaller noise. This phenomenon is called synchronization.

Researchers in the field of control theory have tried to rigorously analyze the concept of synchronization. Jadbabaie et al. [22] examined the heading angles linearly and presented a series of pointless neighbor grids using the locations of the specific agents. It demonstrated that to ensure the synchronization of the system, the graphs have to be mutually linked in a specific identical manner. Afterward, another study [23] used the headings discretely and examined if the same linear system will synchronize. It demonstrated that we could also achieve synchronization if we connect the union of neighbor graphs infinite times.

Contributions: The main offerings of this study are:

- A multi-colonies-based hierarchical structure is designed which divides the UAVs into three distinct and non-overlapping colonies. In each colony, UAVs form hierarchies with one UAV selected as the colony leader.
- This study uses MMACO for the formation of route of a network of UAVs. MMACO limits the pheromone on each trail to ensure early convergence of ACO.
- MMACO finds the best ant of each colony. Then, MAS designates the best ant of each colony as the leader and the rest of the ants as agents.
- All colonies' leaders perform the synchronization and coordination to cooperatively share the mission requirement among all the colonies, and the followers follow their direct leader to achieve the inner group coordination as well the overall mission requirement i.e., path planning of a network.

Organization: This research structured as follows: Section 2 describes the problem and mission scenario. Section 3 sheds some light on the solution framework. Section 4 deals with the model of the proposed system and describes in detail the ACO, MMACO, and Vicsek MAS components of our proposed strategy. Section 5 presents the flowchart and the algorithm. Section 6 discusses the simulation results. Lastly, Section 7 concludes the whole article.

2. Problem Formulations

This section describes the problem that this paper is trying to solve. Firstly, in this research, we simulate the scenario by parallel implementing the design hierarchy with the non-dominated sorting genetic algorithm II (NSGA-II) to check the validity of our scheme. Hereinafter, this research consists of two mission situations, inorder to validate the proposed strategy, i.e., (i) validate the proposed algorithm in a rough environment with hilly peaks. (ii) Crooscheck the execution of the designed model in an environment with tornadoes.

Figure 1 shows the mission scenario in which three colonies have to reach their destination through a dynamic territory. To achieve our objective, we have divided our mission into two main phases. In the first phase, MMACO finds the most optimal route for the leader of each colony. Then, in the second phase, the rest of the two UAVs follow their leader as agents toward the destination using Vicsek MAS. Keep in mind that the three colonies will come closer and closer towards the end and act as one single network. Additionally, there are some obstacles such as mountain peaks and tornados in the mission environment. Due to the aforementioned threats in the area, all UAVs need to coordinate and synchronize to avoid any collision.

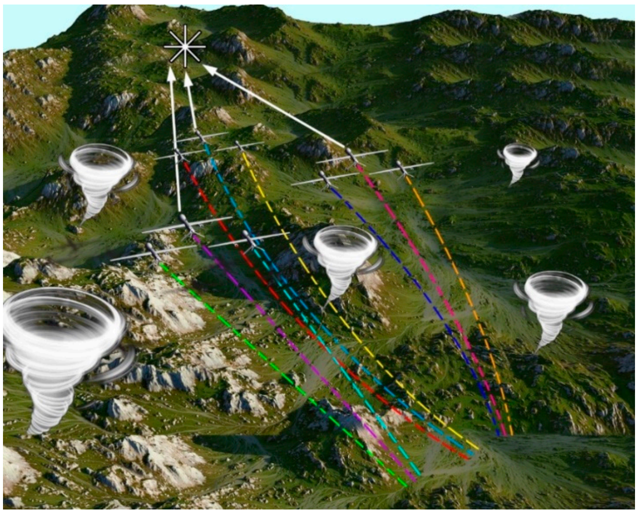


Figure 1. Mission Scenario.

A regular mesh is used as the topology of the network which means that all the edges are of the same length. The hills and tornadoes are modeled as missing nodes. In Figure 2 there is an association in legs, edges and nodes. In order to understand the concept of path planning, it is essential to first understand the concept of nodes, edges, and legs. The nodes are intermediate points plotted by the algorithm between starting position and the final position. The UAV only moves from one node to another. The algorithm plots the nodes for the whole path during one iteration. As the air space is 3D, we split it into three 2D planes, i.e., xy, yz, and zx, to make the computations simpler [24].

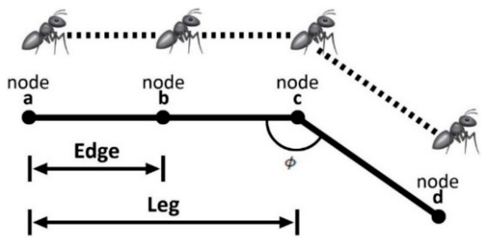


Figure 2. Nodes, Edges, and Legs.

Every time a UAV is set to hit something, the algorithm alters its path and generates a leg. After every turn, a new leg is generated. It is clear from Figure 2 that an edge connects two nodes. For example, edge (a, b) is the length from node a to b. There are 4 nodes, 3 edges, and 2 legs in Figure 2. ϕ is the angle between the legs. The complete consists of multiple legs.

3. Solution Frameworks

This section develops a solution framework to distribute the main objective into smaller and simpler tasks to achieve our mission relatively quickly. Figure 3 illustrates the framework in detail.

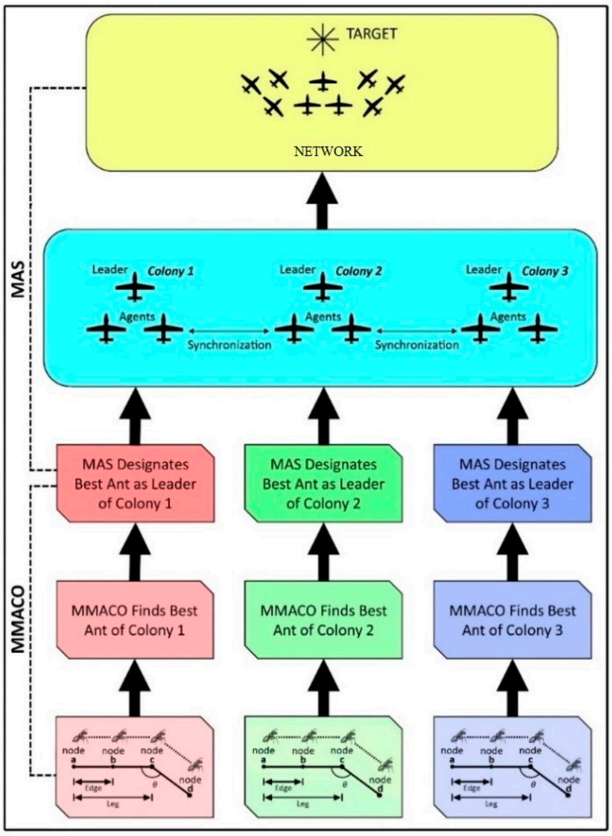


Figure 3. The Solution Framework.

It is clear from Figure 3 that the first task of our system is to plan the path for the leader. Here, we are using MMACO to find the most optimal route while also avoiding obstacles. The concept of path planning is again illustrated here in terms of nodes, edges, and legs. MMACO plots the nodes and the UAV moves from node to node until it encounters an obstacle. Then, it turns and creates a leg until it reaches its destination. MMACO cannot find the best path on the first try so this process is repeated a predetermined number of times. The number of iterations is chosen based on a good estimate of when the algorithm will find the best path.

After MMACO finds the best ant of a colony, the MAS designates it as the leader of that colony and the rest of the ants as its agents. Then, the agents of each colony start to follow their leader. The colonies also synchronize and coordinate with each other to avoid a collision. The colonies get closer and closer as they move towards the destination until they form a network.

The algorithm detects the tornadoes in real-time as a dynamic threat because they are modeled as missing nodes. So, whenever the Vicsek MAS detects a missing node as its neighbor, it changes direction.

4. Model of the Proposed System

In this section of the research article, we design a novel scheme using MMACO to plan the routes of UAVs. Moreover, the proposed scheme will also use Vicsek MAS to coordinate and synchronize all the colonies.

4.1. Ant Colony Optimization

The ant colony optimization (ACO) imitates the actual ants foraging for food using pheromones left over by previous ants [25]. The route traveled most by the ants has the most pheromones, and hence it helps the next ant in selecting the shortest path [26].

The aircraft is flying from the takeoff to the destination using the edges [27]. The following supposition helps select the next edge of the path.

Supposition 1: Assume that the kth ant is located at a node on time t, the transit probability is given as;

$$p^k_{a,b}(t) = \frac{\tau^{\alpha}_{a,b}(t)\mu^{\beta}_{a,b}(t)}{\sum_{b\in accept(a)} \tau^{\alpha}_{a,b}(t)\mu^{\beta}_{a,b}(t)} \tag{1}$$

where $p^k_{a,b}(t)$ is transit probability of the kth ant from node a to node b, $\tau_{a,b}(t)$ is the pheromone on the edge (a, b), $\mu_{a,b}(t)$ is the feasibility of transition from node a to node b, $accept(a)$ is the set of neighboring nodes of a, α is the parameter to control the influence of $\tau_{a,b}(t)$, and β is the parameter to control the influence of $\mu_{a,b}(t)$.

When the algorithm starts, the initial pheromone rate changes with respect to the edges. Afterward, every ant that produced the outcome [28] starts the next cycle of the algorithm and resets the pheromone rate. The pheromone $\tau_{a,b}(t)$ on the edge $(a,\ b)$ is;

$$\tau_{a,b}(t+1) = (1-\rho)\cdot\tau_{a,b}(t) + \sum_{k=1}^{m} \Delta\tau^k_{a,b}(t) \tag{2}$$

where ρ is the evaporation rate of pheromone $(0 \leq \rho \leq 1)$, m is the total number of ants, and $\Delta\tau^k_{a,b}(t)$ is the rate of pheromone on the edge (a,b). $\Delta\tau^k_{a,b}(t)$ can be further defined as

$$\Delta\tau^k_{a,b}(t) = \begin{cases} (Q/L_k) if\ ant\ k\ use\ edges\ of(a,b) \\ (0) \qquad moreover \end{cases} \tag{3}$$

where, Q is the constant, and L_k is the length of the path constructed by the k^{th} ant.

4.2. Max-Min ACO

To ensure early convergence of the ACO, we need to modify the classic algorithm. In this regard, MMACO offers some promising results by limiting the pheromones on each trail [29]. To discuss MMACO, we have to first study the cost of the path. The next formula presents the average path cost $j_{i,m}(t)$;

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

Remark 1. *The k^{th} ant will only update the pheromone if the path cost of the k^{th} ant in the t^{th} iteration fulfill $j_{i,min}(t) \geq j_{i,k}(t)$.*

After each iteration, the algorithm will update the path using Equation (3). MMACO obtains the most optimal and least optimal path after every iteration [30]. It discards the least optimal path to increase the chances of finding the globally best path. Therefore, Equation (3) can be updated as,

$$\Delta\tau^k_{a,b}(t) = \begin{cases} Q/L_b\ , & route\ (a,b)\ refer\ to\ the\ optimize\ route \\ -Q/L_w, & route\ (a,b)\ refer\ to\ the\ worst\ route \\ 0, & otherwise \end{cases} \tag{5}$$

In Equation (5), L_b is the best path and L_w is the least optimal path of the current iteration. MMACO limits the amount of pheromone by restricting it to predetermined levels. This restriction helps accelerate the convergence and avoids falling into stagnation.

Definition 1. *MMACO restricts the pheromone trails to a predetermined maximum and minimum value represented as τ_{max} and τ_{min}. Therefore, by refining the pheromone trails after every cycle, we use this formula to update the pheromone,*

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

4.3. Vicsek MAS

The Vicsek MAS consists of n autonomous agents moving in the plane with the same absolute velocity [31]. The model updates the heading of every agent with respect to the state of its neighbor. The neighbors of an agent I ($1 \leq I \leq n$) at time t are those which lie within a circle of radius r ($r > 0$) centered at the current position of agent i. We represent the neighbors of the agent i at time t as $N_i(t)$,

$$N_i(t) = \{ j | d_{ij}(t) < r \} \tag{7}$$

whereas $d_{ij}(t)$ can be found using the Pythagorean theorem

While, ($x_i(t)$, $y_i(t)$) are the coordinates of the agent at time t. Agent i and agent j are neighbors to each other. Every agent in the system has the same absolute velocity v ($v > 0$).

$$\begin{cases} x_i(t+1) = x_i(t) + v\cos\theta_i(t) \\ y_i(t+1) = y_i(t) + v\sin\theta_i(t) \end{cases} \tag{8}$$

where $\theta_i(t)$ represents the heading angle of agent i at time t. The MAS updates the heading angle using the following equation,

$$\theta_i(t+1) = \arctan\frac{\sum_{j\in N_i(t)} \sin\theta_j(t)}{\sum_{j\in N_i(t)} \cos\theta_j(t)} \tag{9}$$

Equation (9) is used to detect obstacles by checking their neighbors. If there is a missing node in the neighbor (meaning an obstacle), then it will change the heading angle to avoid collision with the obstacle.

The Vicsek MAS model is a dynamic system, and we can use elementary graph theory to analyze this model. Keep in mind that the neighbors of each agent do not remain the same. To define the coordination of agents, Jadbabaie et al. [22] made use of an undirected graph sequence $\mathbb{G}_t = \{\mathcal{V},\ \mathcal{E}_t\}$. Whereas, $\mathcal{V}$ = {1, 2, $\cdots$, N} is the set that contains all agents, and $\mathcal{E}_t$ is the edge set that changes with time. If any two vertexes of a graph are joined together, that graph would be considered connected.

We can simplify Equation (9) as,

$$\tan\theta_i(t+1) = \sum_{j\in N_i(t)} \frac{\cos\theta_j(t)}{\sum_{k\in N_i(t)}\cos\theta_k(t)} \tan\theta_i(t) \tag{10}$$

We can also simplify Equation (10) by using a matrix,

$$\tan\theta(t+1) = A(t)\tan\theta(t) \tag{11}$$

whereas, $\tan\theta(t) \triangleq (\tan\theta_1(t), \ldots, \tan\theta_N(t))^{\tau}$, $A(t) \triangleq \left(a_{ij}(t)\right)$ is the weighted average matrix of the graph $\mathbb{G}_t$:

$$a_{ij}(t) = \begin{cases} \frac{\cos\theta_j(t)}{\sum_{k\in N_i(t)}\cos\theta_k(t)} & if\ (i,\ j) \in \mathcal{E}_t \\ 0, & otherwise \end{cases} \tag{12}$$

To study the synchronization of the Vicsek MAS, Jadbabaie et al. [22] examined the linearized model of Equation (9), which is:

$$\theta_i(t+1) = \frac{1}{n_i(t)} \sum\nolimits_{j \in N_i(t)} \theta_j(t) \tag{13}$$

where, $n_i(t)$ is the number of elements in $N_i(t)$. Similarly, we can replace Equation (11) by,

$$\tan\theta(t+1) = \widetilde{A}(t)\theta(t) \tag{14}$$

where, $\theta(t) \triangleq (\theta_1(t), \ldots, \theta_N(t))^\tau$, and the entries of the matrix $\widetilde{A}(t)$ are,

$$\widetilde{a}_{ij}(t) = \begin{cases} \frac{1}{n_i(t)}, & if\ (i,j) \in \mathcal{E}_t \\ 0, & otherwise \end{cases} \tag{15}$$

4.4. Synchronization and Connectivity

In order to further examine the synchronization of the Vicsek MAS as well as the connectivity of the related neighbor graphs, we have to first define what synchronization is.

Definition 2. *The aforementioned Vicsek MAS model achieves synchronization if the headings of every agent meet the following condition,*

$$\lim_{t\to\infty} \theta_i(t) = \theta,\ i = 1, \ldots, N \tag{16}$$

whereas, θ may vary according to the initial states$\{\theta_i(0),\ x_i(0),\ y_i(0),\ i = 1, \ldots, N\}$and the model constraints v, and r.

The subsequent two propositions will set up the synchronization for the Vicsek MAS and its linear form under the parameters of the initial states and the system constraints r, v, and N.

Proposition 1. *For the Vicsek MAS model in Equations (8) and (9), let $\{\theta_i(0) \in (-\frac{\pi}{2}, \frac{\pi}{2}),$ $i = 1, \ldots, N\}$, and the initial neighbor $\mathbb{G}_0 = \{\mathcal{V},\ \mathcal{E}_0\}$ is connected. So, the model will achieve synchronization if it meets the following condition,*

$$v \leq \frac{d}{\triangle_0}\left(\frac{\cos\overline{\theta}}{N}\right)^N \tag{17}$$

where N is the number of agents, and

$$\begin{cases} \overline{\theta} = \max\limits_i |\theta_i(0)| \\ d = r - \max\limits_{i,j \in \mathcal{E}_0} d_{ij}(0) \\ \triangle_0 = \max\limits_{i,j}\{\tan\theta_i(0) - \tan\theta_j(0)\} \end{cases} \tag{18}$$

Proposition 2. *For the linearized Vicsek model (8) and (13), let $\theta_i(0) \in [0, 2\pi)$, and the initial neighbor graph is connected. So, the model will achieve synchronization if it meets the following condition,*

$$v \leq \frac{d\left(\frac{1}{N}\right)^N}{2\pi} \tag{19}$$

where d is defined as in Equation (18).

5. Flowchart and Designed Algorithm

Figure 4 presents the detailed flowchart of our proposed model. It clearly explains how the two different algorithms work in tandem to achieve our mission objective. The first part of the flowchart pertains to the MMACO and the last part explains the Vicsek MAS. MMACO works with each colony separately and finds its best path. Then, Vicsek MAS synchronizes and coordinates with each colony until the target destination. The flowchart is given below;

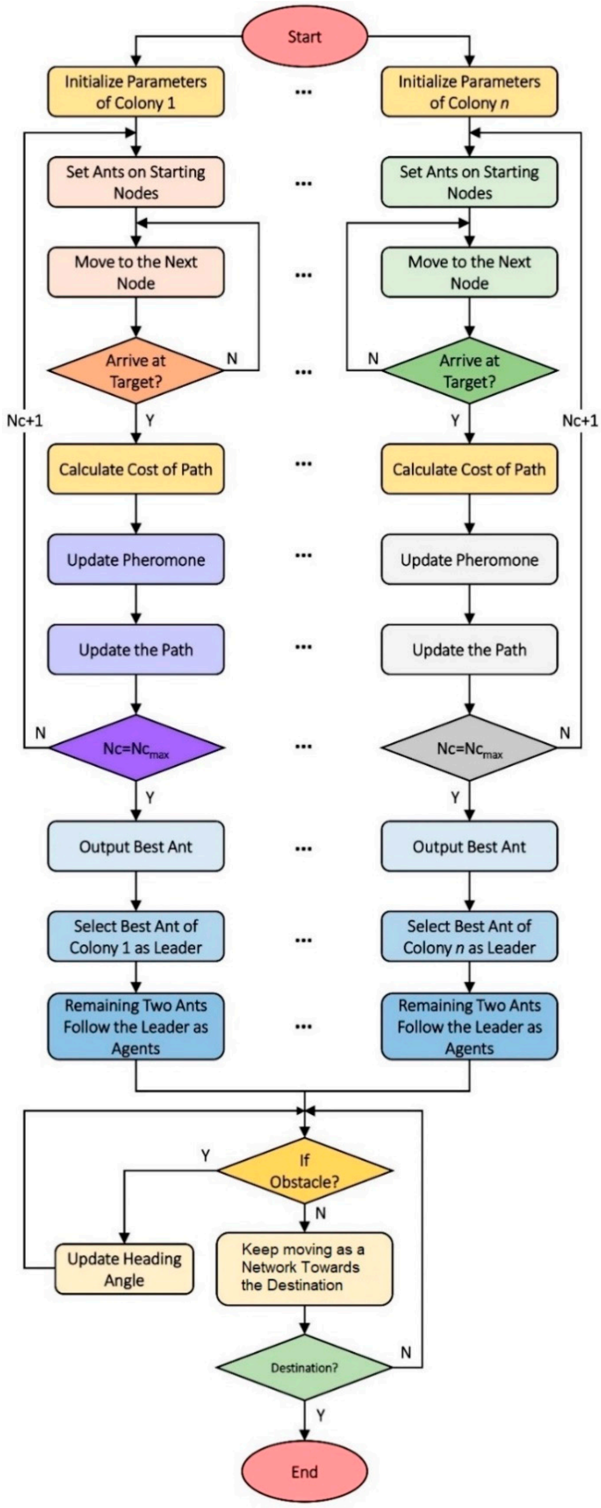


Figure 4. Flowchart of the Proposed Model.

Algorithm 1 of the proposed method is detailed below;

Algorithm 1 Proposed Method

```
Initialize the parameters of the algorithm
procedure algo1
    while(Nc<Ncmax) do
        Place artificial ants on starting position
        If (Reached Destination) then
            Determine average path cost using Equation (4) and update the phenomenon
            using Equations (2), (3) and (5)
            else
            Move to the next node using Equation (1)
        end if
        Output the best ant
    end while
    while (destination) do
        if (Found_Obstacle) then
            Update heading angle using Equation (9)
            else
            Keep moving towards the destination
        end if
    end while
end procedure
```

6. Simulation Results

The efficiency and validation of the proposed strategy is defined in this section. By simulating the scenario using MATLAB 2019. The dimensions for the mission area are $(x, y, z) = (30, 30, 2)$ km. Firstly, in this research, we simulate the scenario by parallel implementing the design hierarchy with the non-dominated sorting genetic algorithm II (NSGA-II) algorithm to check the validity of our scheme.

Table 1 presents the constraints used in the algorithm. Now α and β control the effect of pheromone or heuristics on the outcome. To get a heuristic solution, we select a bigger value of β compared to α. The pheromone evaporation rate is ρ. ρ controls the speed at which the pheromones evaporate from the path. We want a relatively higher value for ρ because we need the previous pheromones to evaporate faster and to make way for new pheromones with better information. Nc_{max} represents the maximum number of iterations, m is the total ants. Nc_{max} and m depend on each other. The higher the number of ants, the lower the iterations. If we decreased the number of ants we need to run more iterations.

Table 1. Constraints for the Algorithm.

Constraint	Symbol	Value
Pheromone constraint	α	3
Heuristics constraint	β	4
Pheromone evaporation rate	ρ	0.8
Max iterations	Nc_{max}	50
No. of ants	m	20
Constant that affects the pheromone rate	Q	15
Min pheromone	τ_{min}	0.1
Max pheromone	τ_{max}	1
Min velocity	v_{min}	40 ms^{-1}
Max velocity	v_{max}	70 ms^{-1}
Max turning angle	ϕ_{max}	105°

Q is the constant that controls the pheromone rate. We have selected a relatively higher value of Q because we want more pheromone depositing on the trail. τ_{min} and τ_{max} are the minimum and maximum pheromones values. The reason we restricted the algorithm within this interval is that it has the best chance of discovering the best solution. The minimum and maximum velocities of the UAV are v_{min} and v_{max}. These velocities are chosen to best represent the real-life speeds of the UAVs. Lastly, ϕ_{max} is the maximum turning angle. The reason to choose 105° is that at higher angles, chances of collision among UAVs increase.

First, we compare our proposed method with the NSGA-II algorithm to prove the effectiveness of our method. NSGA-II is a multi-objective evolutionary algorithm. It is an efficient algorithm and works on a non-dominated sorting method. It uses an elitist strategy and has an effective parameter handling approach.

Figure 5 presents the comparison between the two methods. As it is clear from the plot, our proposed method finds a shorter path than the NSGA-II while also avoiding the hilly peaks and tornadoes. This is because our proposed method is computationally faster than the NSGA-II and hence, can plot the path quicker than NSGA-II while avoiding the obstacles. Meanwhile, the NSGA-II also takes more turns than our proposed method. Frequent turns can introduce instability and inconsistency in the flight.

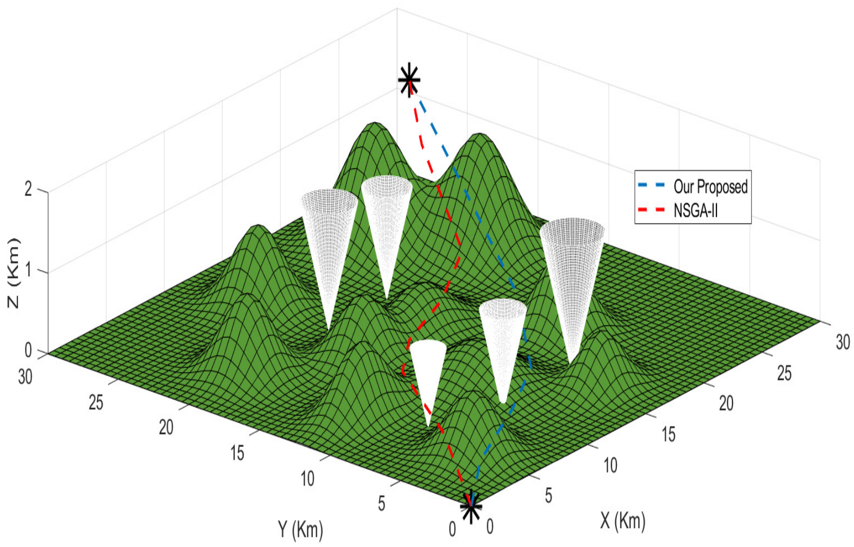


Figure 5. Comparison of the proposed algorithm with NSGA-II algorithm.

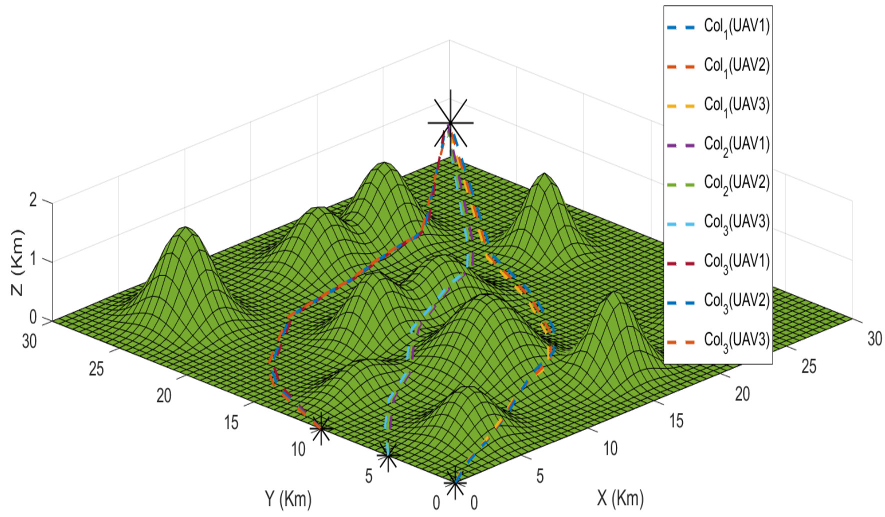
Table 2 presents the flight details of the comparison between the proposed method and the NSGA-II algorithm. Confirming the previous simulation given in Figure 5, we can see that the path planned by the proposed method is over 2 km shorter than the NSGA-II. This proves the superiority of the designed strategy over NSGA-II.

Table 2. Flight Details of proposed algorithm and NSGA-II.

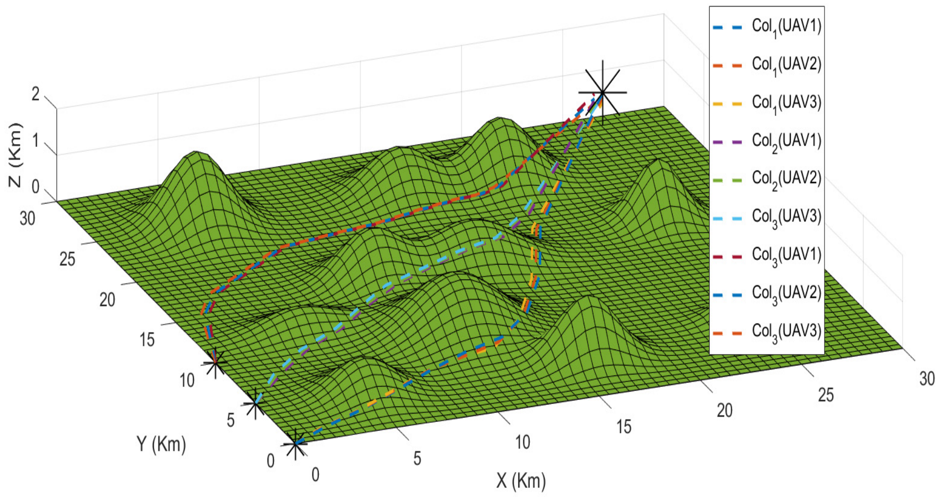
	Initial Position	Destination	Distance
Proposed Algorithm	(0,0,0)	(25,25,1.8)	34.321
NSGA-II	(0,0,0)	(24.8,24.8,1.81)	36.393

Case Study I: In this scenario, to test the efficiency of the designed strategy, we consider three ant colonies that launch from three different positions and arrive at the destination concurrently. The environment is hazardous and contains hilly peaks. First, each colony uses MMACO to find its best ant. MMACO achieves this by continuously modeling a journey of a colony of ants from the initial point to the destination. Whenever an ant finds an obstacle, it changes its direction and leaves a pheromone for other ants so they can avoid the obstacle as well. This process is repeated a fixed number of times until the best ant with the shortest distance is discovered.

Afterward, Vicsek MAS designates the best ant as the leader and the rest of the ants follow the leader as agents. Figure 6 illustrates the proposed strategy in action and shows that our strategy will easily fulfill the mission requirement. Figure 6a presents the 3D simulation from one view while Figure 6b presents it from another angle.



(a)



(b)

Figure 6. Simulation Results with Hilly Peaks; (**a**) 3D simulation seen from one point of view (**b**) 3D simulation from another point of view.

Figure 7 demonstrates the optimization costs of finding the best path. It is clear from Figure 7 that we find no significant decline in the cost after the 5th iteration. There is a slight variation in the costs of colonies 2 and 3 but is negligible. The reason for this decline is that the algorithm has discovered the best route around the 5th iteration. Afterward, the ants just travel on the best path and hence almost the same cost for the subsequent iterations.

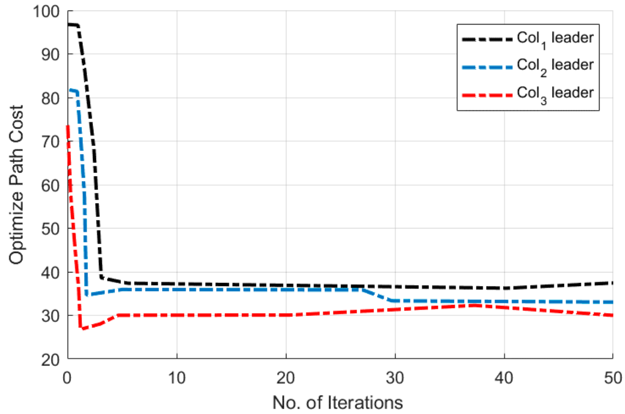


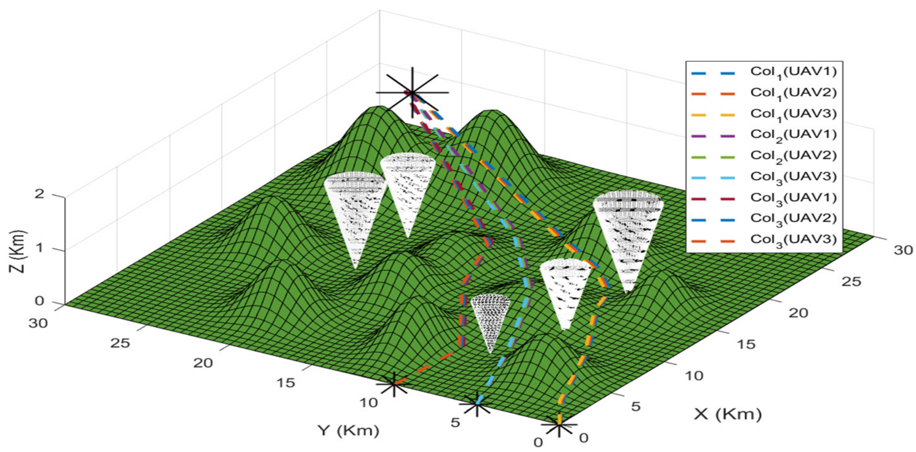
Figure 7. Optimization Costs for Case Study I.

Table 3 shows the flight details of the best ant of each colony. Table 3 also shows the complete flight distance from the initial position to the target.

Table 3. Flight Details of Best Ants of Each Colony for Case I.

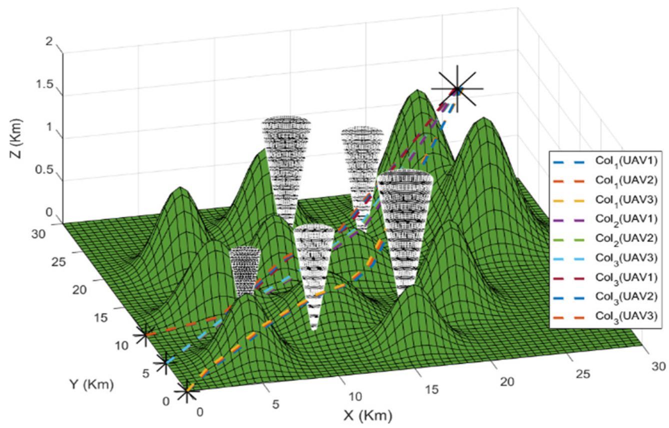
	Initial Position	Destination	Distance
Col1 best ant	(0,0,0)	(25,25,1.3)	37.379
Col2 best ant	(0,5,0)	(24.8,25,1.3)	33.386
Col3 best ant	(0,10,0)	(24.6,25,1.3)	30.092

Case Study II: In this scenario, to test the efficiency of the designed strategy, we consider three ant colonies (UAVs) that launch from three different positions and arrive at the destination concurrently. The environment is dynamic and contains not only hilly peaks but also tornadoes. First, each colony uses MMACO to find its best ant. The process of finding the ant is similar to the one explained in *Case Study I*. Afterward, Vicsek MAS designates the best ant as the leader and the rest of the ants follow the leader as agents. Figure 8 illustrates the proposed strategy in action and shows that our strategy will easily fulfill the mission requirement. Figure 8a shows the mission area from one point of view while Figure 8b presents it with another view to clearly show the peaks and tornadoes.



(**a**)

Figure 8. *Cont.*



(b)

Figure 8. Simulation Results with Peaks and Tornadoes; (**a**) 3D simulation seen from one point of view (**b**) 3D simulation from another point of view.

Figure 9 demonstrates the optimization costs of finding the best path. It is clear from Figure 9 that we find no significant decline in the cost after the 10th iteration. The reason for this decline is that the algorithm has discovered the best route at around the 10th iteration. Afterward, the ants just travel on the best path and hence almost the same cost for the subsequent iterations.

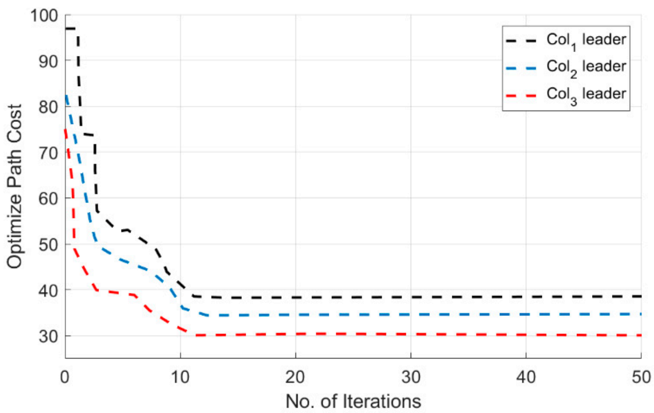


Figure 9. Optimization Costs for Case Study II.

Table 4 shows the flight details of the best ant of each colony. Table 4 also shows the complete flight distance from the initial position to the target.

Table 4. Flight Details of Best Ants of Each Colony for Case II.

	Initial Position	Destination	Distance
Col1 best ant	(0,0,0)	(25,25,1.46)	37.385
Col2 best ant	(0,5,0)	(24.8,25,1.46)	33.393
Col3 best ant	(0,10,0)	(24.6,25,1.46)	30.099

7. Conclusions

This paper presented a bio-inspired strategy for the control and path planning of UAVs that make a network under a rough and hazardous environment. The UAVs are prearranged in different and non-overlapped colonies, which formed a hierarchy in the individual colony. The UAVs in the colonies are divided into the leader and its follower.

To solve the predefined mission scenario, the paper combined Max-Min ant colony optimization (ACO) with Vicsek based multi-agent system (MAS) to make a meta-heuristic algorithm. Moreover, in our designed control structure, Max-Min (ACO) is used to bound, lemmatize the pheromone, and gives the best ants to every colony, which makes the most optimized path. Lastly, we tested the proposed method in two different dynamic environments, and the simulation results demonstrated that the proposed method is robust and efficient. The designed strategy finds the best path quickly with minimum cost. The simulation results also compared the proposed method with the NSGA-II algorithm, to prove the designed algorithm has a better convergence rate and optimal path. Future work includes increasing the number of UAVs in each colony and also increasing the number of colonies.

Author Contributions: This conceptualization and methodology of this research was done by Z.A.A., E.H.A. and M.S. confirms the validation, revised the article and writing. The final proof read was done by Z.A.A. and E.H.A. All authors have read and agreed to the published version of the manuscript.

Funding: The initial phase of this research was supported by the startup grant of Beijing Normal University, Zhuhai, China and the validation funding was supported by Taif University Researchers Supporting Project number (TURSP-2020/292) Taif University, Taif, Saudi Arabia.

Data Availability Statement: All the data used to support the findings of this research are included in this article.

Acknowledgments: The authors would like to acknowledge Taif University Researchers Supporting Project number (TURSP-2020/292) Taif University, Taif, Saudi Arabia.

Conflicts of Interest: There is no conflict of interest between the authors.

References

1. Jain, G.; Yadav, G.; Prakash, D.; Shukla, A.; Tiwari, R. MVO-based path planning scheme with coordination of UAVs in 3-D environment. *J. Comput. Sci.* **2019**, *37*, 101016. [CrossRef]
2. Hayyolalam, V.; Kazem, A.A.P. Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103249. [CrossRef]
3. Fouad, M.M.; El-Desouky, A.I.; Al-Hajj, R.; El-Kenawy, E.S.M. Dynamic group-based cooperative op-timization algorithm. *IEEE Access* **2020**, *8*, 148378–148403. [CrossRef]
4. Song, S.; Choi, M.; Goh, Y.; Yun, J.; Yoo, W.; Yang, W.; Jung, J.; Chung, J.-M. Analysis of wireless backhaul networks based on aerial platform technology for 6g systems. *Comput. Mater. Contin.* **2020**, *62*, 473–494. [CrossRef]
5. Wang, J.; Gao, Y.; Zhou, C.; Sherratt, S.; Wang, L. Optimal coverage multi-path scheduling scheme with multiple mobile sinks for WSNs. *Comput. Mater. Contin.* **2020**, *62*, 695–711. [CrossRef]
6. Wang, L. Optimize 2-D path planning of mobile robot by using ACO with guidance factor. *Sir Syed Univ. Res. J. Eng. Technol.* **2019**, *9*. [CrossRef]
7. Zhao, W.; Wang, L.; Zhang, Z. Supply-demand-based optimization: A novel economics-inspired algo-rithm for global optimization. *IEEE Access* **2019**, *7*, 73182–73206. [CrossRef]
8. Hamed, A.Y.; Alkinani, M.H.; Hassan, M.R. Ant colony optimization for multi-objective multicast routing. *Comput. Mater. Contin.* **2020**, *63*, 1159–1173. [CrossRef]
9. Liu, J.; Xu, S.; Zhang, F.; Wang, L. A hybrid genetic-ant colony optimization algorithm for the optimal path selection. *Intell. Autom. Soft Comput.* **2017**, *23*, 235–242. [CrossRef]
10. Cao, X.; Zhu, D. Multi-AUV task assignment and path planning with ocean current based on biological inspired self-organizing map and velocity synthesis algorithm. *Intell. Autom. Soft Comput.* **2015**, *23*, 31–39. [CrossRef]
11. Al-Shihabi, S.; AlDurgam, M. Multi-objective optimization for the multi-mode finance-based project scheduling problem. *Front. Eng. Manag.* **2020**, *7*, 223–237. [CrossRef]
12. Caruso, G.; Gattone, S.A.; Balzanella, A.; Di Battista, T. Cluster analysis: An application to a real mixed-type data set. In *Models and Theories in Social Systems*; Springer International Publishing: Cham, Switzerland, 2019; pp. 525–533.
13. Arora, S.; Singh, H.; Sharma, M.; Sharma, S.; Anand, P. A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection. *IEEE Access* **2019**, *7*, 26343–26361. [CrossRef]
14. Dhiman, G.; Kaur, A. A hybrid algorithm based on particle swarm and spotted hyena optimizer for global optimi-zation. In *Soft Computing for Problem Solving*; Springer International Publishing: Singapore, 2019; pp. 599–615.
15. Luan, J.; Yao, Z.; Zhao, F.; Song, X. A novel method to solve supplier selection problem: Hybrid algorithm of genetic algorithm and ant colony optimization. *Math. Comput. Simul.* **2019**, *156*, 294–309. [CrossRef]
16. Stützle, T.; Hoos, H.H. Max–Min ant system. *Future Gener. Comput. Syst.* **2000**, *16*, 889–914. [CrossRef]

17. Tran, V.P.; Garratt, M.; Petersen, I.R. Switching time-invariant formation control of a collaborative multi-agent system using negative imaginary systems theory. *Control Eng. Pract.* **2020**, *95*, 104245. [CrossRef]
18. Zeghida, D.; Meslati, D.; Bounour, N. Bio-IR-M: A multi-paradigm modelling for bio-inspired multi-agent systems. *Informatica* **2018**, *42*, 451–466. [CrossRef]
19. Tian, B.; Zuo, Z.; Wang, H. Leader-follower fixed-time consensus of multi-agent systems with high-order inte-grator dynamics. *Int. J. Control* **2017**, *90*, 1420–1427. [CrossRef]
20. Wang, X.; Li, S.; Yu, X.; Yang, J. Distributed active anti-disturbance consensus for leader-follower higher-order multi-agent systems with mismatched disturbances. *IEEE Trans. Autom. Control* **2016**, *62*, 5795–5801. [CrossRef]
21. Vicsek, T.; Czirok, 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*, 1226–1229. [CrossRef] [PubMed]
22. Jadbabaie, A.; Lin, J.; Morse, A. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Autom. Control* **2003**, *48*, 988–1001. [CrossRef]
23. Savkin, A.V. Coordinated collective motion of groups of autonomous mobile robots: Analysis of Vicsek's model. *IEEE Trans. Autom. Control* **2004**, *49*, 981–982. [CrossRef]
24. Duan, H.B.; Zhang, X.Y.; Wu, J.; Ma, G.J. Max-min adaptive ant colony optimization approach to multi-UAVs coordinated trajectory replanning in dynamic and uncer-tain environments. *J. Bionic Eng.* **2009**, *6*, 161–173. [CrossRef]
25. Gutjahr, W.J. ACO algorithms with guaranteed convergence to the optimal solution. *Inf. Process. Lett.* **2002**, *82*, 145–153. [CrossRef]
26. Dissanayake, S.D.; Armstrong, J. Comparison of aco-ofdm, dco-ofdm and ado-ofdm in im/dd systems. *J. Light. Technol.* **2013**, *31*, 1063–1072. [CrossRef]
27. Yang, X.J.; Huang, Z.G. Opposition-based artificial bee colony with dynamic cauchy mutation for function optimization. *Int. J. Adv. Comput. Technol.* **2012**, *4*, 56–62.
28. Dorigo, M.; Di Caro, G. Ant colony optimization: A new meta-heuristic. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; IEEE: Piscataway, NJ, USA, 1999; Volume 2, pp. 1470–1477.
29. Ali, Z.A.; Zhangang, H.; Zhengru, D. Path planning of multiple UAVs using MMACO and DE algorithm in dynamic environment. *Meas. Control* **2020**. [CrossRef]
30. Dorigo, M.; Socha, K. An introduction to ant colony optimization. In *Handbook of Approximation Algorithms and Metaheuristics*, 2nd ed.; Chapman and Hall/CRC: Boca Raton, FL, USA, 2006; pp. 395–408.
31. Liu, Z.; Guo, L. Connectivity and synchronization of Vicsek model. *Sci. China Ser. F Inf. Sci.* **2008**, *51*, 848–858. [CrossRef]