Equilibrium Optimizer with Generalized Opposition-based Learning for Multiple Unmanned Aerial Vehicles Path Planning

Yang Chen (✉ shawn.cy@nuaa.edu.cn)

Nanjing University of Aeronautics and Astronautics <https://orcid.org/0000-0001-5239-9816>

Dechang Pi

Nanjing University of Aeronautics and Astronautics

Bi Wang

Southeast University

Ali Wagdy Mohamed

Cairo University

Junfu Chen

Nanjing University of Aeronautics and Astronautics

Research Article

Keywords: Equilibrium optimizer, Multi-UAVs path planning, Global optimization, Numerical optimization

Posted Date: December 1st, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-1125211/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Equilibrium Optimizer with Generalized Opposition-based Learning for Multiple Unmanned Aerial Vehicles Path Planning

Yang Chen^a, Dechang Pi^a, Bi Wang^b, Ali Wagdy Mohamed^{c,d}, Junfu Chen^a

^a*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China*

^b*School of Computer Science and Engineering, Southeast University, Nanjing, 211189, China*

^c*Department of Operations Research, Faculty of Graduate studies for Statistical Research, Cairo University, Giza, 12613, Egypt.*

^d*Department of Mathematics and Actuarial Science, School of Sciences & Engineering, The American University in Cairo, Cairo, 11435, Egypt*

Abstract

Multiple Unmanned Aerial Vehicles (UAVs) path planning is the benchmark problem of multiple UAVs application, which belongs to the non-deterministic polynomial problem. Its objective is to require multiple UAVs flying safely to the goal position according to their specific start position in three-dimensional space. This issue can be defined as a high-dimensional optimization problem, the solution of which requires optimization techniques with global optimization capabilities. Equilibrium optimizer (EO) is a population-based meta-heuristic algorithm. In order to improve the optimization ability of EO to solve high-dimensional problems, this paper proposes a modified equilibrium optimizer with generalized opposition-based learning (MGEO), which improves the population activity by increasing the internal mutation and cross of the population. In addition, the generalized opposition-based learning is used to construct the population, which can effectively ensure that the algorithm has ability to jump out of the limitation of local optimal. Firstly, numerical experiments show that

*Corresponding author

Email addresses: shawn.cy@nuaa.edu.cn (Yang Chen), dcpi@nuaa.edu.cn (Dechang Pi), wangbi@seu.edu.cn (Bi Wang), aliwagdy@gmail.com (Ali Wagdy Mohamed), cjf@nuaa.edu.cn (Junfu Chen)

MGOEO has better optimization precision than EO and several other swarm intelligent algorithms. Then, the paths of UAVs are simulated in three different obstacle environments. The simulation results show that MGOEO can obtain safe and smooth paths, which are better than EO and other eight state-of-the-art optimization algorithms.

Keywords: Equilibrium optimizer; Multi-UAVs path planning; Global optimization; Numerical optimization

1. Introduction

Unmanned Aerial Vehicle (UAV), as a kind of modern equipment with strong mobility and **excellent autonomous control capability**, has received extensive attention from scholars in various research fields. Its research and application have
5 been involved in post disaster medical assistance, intelligent transportation, agricultural science, location monitoring and other modern industries (Yao et al., 2019; Liu et al., 2020; Radoglou-Grammatikis et al., 2020; Sarçiçek & Akkuş, 2015). In the research of UAV application, the problem of UAV flight path planning is the benchmark problem for all applications and is the precondition to
10 ensure that UAV can service better. The issue of UAV flight path planning can be described as requiring the UAV to get an optimal path out of all the paths from the starting point to the target position, with minimal cost under conditions that require avoiding all obstacles and maintaining steady flight. There have been many achievements in the research of a single UAV from the two-
15 dimensional environment to three-dimensional environment (Lin et al., 2019; Qu et al., 2020; Yao & Wang, 2017). These researchers focus on adding complex constraints to the flight environment to explore the safe path, such as increasing the number and coverage range of radar and artillery in the simulated battlefield environment. However, **as research evolves and the needs of industry become**
20 **more complex, the collaborative application of multi-UAVs will become a focus of future research.**

The problem with multi-UAVs path planning is that multiple vehicles are

required to complete their respective missions with different paths on the basis of a single UAV. The issue can be regarded as a high-dimensional optimization problem. The traditional algorithms A* and D* are often used for solving the land robots path planning, and their improved versions are also used for UAV path planning (Guruji et al., 2016; Cai et al., 2019; Guo et al., 2009). However, the solutions often obtained are not very satisfactory. Swarm intelligence optimization is often used to solve high-dimensional optimization problems, and it often performs beautifully due to the fact that it can somewhat jump out of the limitations of local optimality (Chen & Pi, 2020). Therefore, researchers apply such algorithms to solving the UAV path planning problem, often resulting in a satisfactory path. Wang et al. used glowworm swarm optimization (GSO) to simulate the flight trajectories of single UAV in two and three-dimensional environments, respectively (Wang et al., 2015). Wang et al. concluded that GSO has a better ability to acquire flight trajectories compared to Dijkstra (Zhang et al., 2017), particle swarm optimization (PSO) (Liu et al., 2018), biogeography-based optimization (BBO) (Simon, 2008), and an improved version of bat algorithm (IBA) (Wang et al., 2016). Therefore, it can be seen that different optimization algorithms have different search capabilities. The path planning of multiple UAVs is a NP hard problem, which requires the algorithm with strong global optimization ability to plan the path for saving total costs.

The idea of swarm intelligence algorithm is to simulate the living foraging behavior of natural social creatures, and searching in the space of objective optimization problem with population as unit, which is also called natural heuristic algorithm. What limits the search capabilities of the swarm intelligence algorithm is the problem of premature convergence in its late search. Different algorithms have been proposed to curb early convergence from different perspectives to enhance the global search capability. These algorithms include multi-Verse Optimizer (MVO) inspired by multi verse theory (Abd Elaziz et al., 2019), whale optimization algorithm (WOA) (Chen et al., 2019a) and salp swarm algorithm (SSA) (Chen et al., 2020) to simulate the behavior of marine organisms, sine-cosine algorithm (SCA) (Gupta & Deep, 2019) based on mathematical functions,

and flower pollination algorithm (FPA) (Salgotra & Singh, 2017), etc. These
55 algorithms have achieved good results in wind prediction, parameter identification and large-scale production scheduling (Chen & Pi, 2019; Chen et al., 2019b; Shao et al., 2020, 2019). At present, researchers apply grey wolf optimizer (GWO) (Dewangan et al., 2019), MVO (Jain et al., 2019) and SSA (Saxena et al., 2019) to optimize the path of multi-UAVs, where SSA is the most efficient.
60 Wolpert et al. pointed out that there is no one optimization algorithm suitable for solving all optimization problems, meaning that any one algorithm is somewhat different in solving a particular optimization problem (Wolpert & Macready, 1997). This paper extends the study of a population-based algorithm called Equilibrium optimizer (EO) that was proposed recently with the idea of
65 the mass balance equation of control volume (Faramarzi et al., 2020). EO has been successfully used for image segmentation (Abdel-Basset et al., 2020).

To improve the optimization performance of EO, we propose a new equilibrium optimizer with generalized opposition-based learning, which is used for function optimization and multi-UAVs path planning. Generalized opposition-based learning is an improved version of the traditional opposition-based learning.
70 The traditional version has been successfully applied in the field of optimization algorithms. Many improvements have emerged in the hairsplitting process, which are aimed at improving the learning rate of the optimization algorithm (Tizhoosh & Ventresca, 2008; Mahdavi et al., 2018; Wang et al., 2011; Draa, 2015; Abd Elaziz et al., 2017; Sapre & Mini, 2019). Compared with the
75 traditional version, the generalized opposition-based learning is different while it uses random numbers to control the changes of the opposition learning points. Therefore, it has a better stochastic learning rate. In addition to embedding generalized opposition-based learning into EO, we add crossover and mutation
80 under random probability in the EO framework, which can increase the diversity of populations and make the algorithm have better ability to jump out of local optimal solutions on high-dimensional problems. The specific MGOEO design is described in section 4 of this paper. First, the performance of the proposed MGOEO is verified by using different types of continuous optimization

85 test functions. Then, MGOEO is applied to multi-UAVs path planning in three different scenarios. The results show that the MGOEO is effective and efficient.

The remainder of this article is organized as follows. Section 2 describes the problem of the multi-UAVs path planning and gives the relevant model definition; Section 3 describes the EO's framework; Section 4 describes the design ideas of the proposed MGOEO and the implementation process; Section 5 carries out numerical experiment test and discusses the results; Section 6 uses the proposed algorithm for multi-UAVs path simulation; Section 7 summarizes the work of this article and gives the future research direction.

2. Problem description

95 The problem of multi-UAVs flight path planning is to plan the tracks of at least three UAVs simultaneously in search space. Its purpose is to require each UAV to take off from the set take-off point and reach the target position safely. As shown in Fig.1, three UAVs A, B and C set their respective target positions respectively. During the course of navigation, they need to avoid the obstacles in the diagram to maintain a smooth flight to the goal point. For UAV i , we

Figure 1: The diagram of the UAV path planning

100

consider the path of the UAV to be l_i , then the objective problem is to get the

combined optimal path of n UAV. Thus the objective function is shown in Eq.1, where c indicates the cost of the UAV's chosen path. The constraint condition means that two UAVs cannot simultaneously choose the same position at the
105 same time when flying in a specific search space.

$$\begin{aligned} f(l_k) &= \arg \min \sum_{k=1}^n c_k l_k, k = 1, 2, \dots, n \\ \text{s.t. } \chi_{i,j}(l_i, l_j) &= 0, \forall i, j = 1, 2, \dots, n \end{aligned} \quad (1)$$

In the process of flight, UAVs need the calculation of flight cost, which includes the loss of fuel consumption, avoiding the consumption of turning obstacles, and the consumption of inconsistent final goal points. For a particular UAV in three-dimensional space, the flight position set from takeoff point s to
110 target point t is recorded as $l_{x,y,z}$, which is shown in Eq.2.

$$l_{x,y,z} = \{s, p_1, p_2, \dots, p_n, t\} \quad (2)$$

Among them, p_1 to p_n is the docking points selected during the flight. Assuming that the UAV flies at a constant speed during the flight, its fuel consumption is directly proportional to the flight distance, so the fuel consumption loss can be expressed as c_{fuel} , and its calculation method is shown in Eq.3

$$c_{fuel} = \sum l_{x,y,z} \quad (3)$$

115 More sharp turns during a UAV flight can lead to more fuel consumption and danger for the UAV. Therefore, sharp turns should be avoided as much as possible in the flight path. So the cost function of a sharp turn represents the minimum number of turns when avoiding an obstacle. In path planning, three consecutive points are taken p_1, p_2, p_3 , where the cross product between vectors
120 $\overrightarrow{p_1 p_2}$ and $\overrightarrow{p_2 p_3}$ is zero, and there is no sharp turn. All others are considered to have turning consumption.

The number of sharp turns is recorded as: $c_{turn} = \text{Count of turning points in path}$.

Considering whether all the UAVs will eventually reach their target locations.
125 Therefore, there will be an approximate range of objective precision for UAVs

in practical applications, which is calculated by Eq.4. The symbol x_{end} , y_{end} , z_{end} respectively represents the actual location of the UAVs in the mission. The symbol x_t , y_t , z_t indicates the specific goal location. Therefore, c_{end} is 0 when the goal is completely overlapped, which is responsible for defining the cost of the objective by distance.

$$c_{end} = \sqrt{(x_{end} - x_t)^2 + (y_{end} - y_t)^2 + (z_{end} - z_t)^2} \quad (4)$$

Therefore, the overall cost function is shown in Eq.5, where the α_1 , α_2 , α_3 respectively represents the weight of fuel consumption cost, sharp turn cost, terminal uncertainty cost, and different values of these parameters indicate different cost priorities.

$$C_{total} = \alpha_1 \cdot C_{fuel} + \alpha_2 \cdot C_{turn} + \alpha_3 \cdot C_{end} \quad (5)$$

3. Equilibrium Optimizer

The idea of equilibrium optimizer is derived from the mass balance equation in the control volume, which can be expressed by the first order ordinary differential equation as Eq.6.

$$V \frac{dC}{dt} = QC_{eq} - QC + G \quad (6)$$

C is the concentration in the control volume V . In engineering thermodynamics, control volume is also called open system, which is called control volume for short. $V \frac{dC}{dt}$ represents the rate of mass change in the control volume. Q Indicates the volume flow in and out of the control volume. C_{eq} is the concentration in equilibrium pool. G is the mass production rate in the control volume.

The model of equilibrium optimizer still evolves in the form of population. An equilibrium pool of five individual components is constructed in the algorithm to provide reference learning individuals. $C_1 - C_4$ represent the top 4 better solution vectors obtained from the population solving the objective problem, and C_{ave} is the arithmetic mean vector of these 4 vectors. The equilibrium pool

is expressed as Eq.7.

$$C_{eq} = \{C_1, C_2, C_3, C_4, C_{ave}\} \quad (7)$$

150 The core update equation for this algorithm is as follows:

$$C = C_{eq} + (C - C_{eq}) \cdot F + \frac{G}{\lambda V}(1 - F) \quad (8)$$

C indicates the position of the individual in Eq.8. C_{eq} is a randomly selected learning object from the equilibrium pool. F is the parameter (turnover rate) that changes over time and is calculated as shown in Eq.9. The λ and r are random numbers between 0 and 1. a_1 is defined as a constant 2.

$$F = a_1 \text{sign}(r - 0.5)(e^{-\lambda t} - 1) \quad (9)$$

155 The G is called the generation rate in Eq.8 and is a condition of the algorithm to increase the development of the exact solution. GCP shown in Eq.10 and Eq.11 is the parameter controlling the generation rate, where r_1, r_2 are all expressed as random numbers between 0 and 1.

$$G = GCP(C_{eq} - \lambda C) \cdot F \quad (10)$$

$$GCP = \begin{cases} 0.5r_1, r_2 \geq 0.5 \\ 0 & r_2 < 0.5 \end{cases} \quad (11)$$

160 3.1. Proposed algorithm

EO algorithm is also a search technology based on population evolution. Generally speaking, other evolutionary strategies of swarm intelligence based on population are elite retention. That is to say, the population moves around the position of the best individual in the current population, or is constrained by the current optimal individual. For example, in PSO, each individual in the population receives the common guidance of each individual's historical optimal solution and the swarm's optimal solution. In global optimization of FPA, populations are exploratory learning toward the optimal position of the current

Algorithm 1 : EO for the Minimization problem

Input: $Maxgen, NP$ **Output:** $Gbest$

```
1: while ( $t < Maxgen$ ) do
2:   for  $i=1:NP$  do
3:     Calculate the fitness of  $C_i$ ;
4:   end for
5:    $C = \text{sort}(C)$ 
6:   Save  $C_1-C_4$  and their arithmetic mean to the equalization pool;
7:   for  $i=1:NP$  do
8:     Execute the relevant equation to calculate the new population position, expressed as  $NewC$  ;
9:     Calculate the fitness of  $NewC_i$ ;
10:    if  $NewC_i < C_i$  then
11:       $C_i = NewC_i$ 
12:    end if
13:  end for
14:   $Gbest = \min(C)$ 
15: end while
```

population. In the EO design, instead of directly using the population-optimal
 170 guideline population search, a balanced selection pool strategy is designed in
 which the best four individuals of the population are retained and the average
 of these four individuals is set as a virtual candidate. In the process of evolution,
 each individual in the population interacts with an individual selected randomly
 from the equilibrium pool to retain a better solution. We can see that this
 175 pattern does not learn directly from the optimal individuals of the population,
 which can effectively delay the convergence of the population. But there are
 still two problems:

1. Balancing global exploration capacity of EO with the local excavation
 capacity becomes weak, and the local exploration capacity of the population
 180 cannot be guaranteed during the late evolution of EO.
2. As the algorithm randomly selects individuals in the equilibrium pool, the
 possibility of learning from weak individuals arises, which increases the global
 search power but slows down the convergence speed of EO in dealing with high
 dimensional problems.

185 These two problems can directly lead to the algorithm's lack of precision
 in solving high-dimensional optimization problems. In order to improve the
 algorithm's search capability and curb its premature fall into local optimal pre-
 mature convergence, we propose an improved version algorithm MGOEO.

Firstly, the idea of cross-mutation is embedded in EO. Cross-mutation is
 190 the main means to update the new solution in evolutionary optimization, using
 cross-mutation can improve the internal diversity of the population. Thus Eq.12
 is used to generate the mutation solution, which is cross-synthesized with the
 mutation solution generated by Eq.8 into new populations for searching.

$$C_i = C_i + r \cdot (C_j - C_k) \quad (12)$$

In addition, opposition-based learning is applied to extending the coverage space
 195 of the population, which can enhance the activity of the population. Opposition-
 based learning (Tizhoosh & Ventresca, 2008) can be described as learning from
 opposing points towards the current point in the objective space. Defining the

opposing point in D-dimensional space as shown in Eq.13. U and L indicate the upper and lower limits of the position respectively. For the position $C_d \in [U_d, L_d]$, where $d = \{1, 2, \dots, D\}$, the opposing point is $\tilde{C}_d$.

$$\tilde{C}_d = U_d + L_d - C_d \quad (13)$$

This was the original version of the opposition-based learning. After a follow-up study, the researchers proposed super opposite and quasi opposite. From a probabilistic point of view, they have a higher probability of searching to a more optimal solution than the original version (Tizhoosh & Ventresca, 2008). Generalized opposition-based learning is a more random approach to learning proposed on this basis. As shown in Eq.14, the generalized opposite point of position $C_d \in [U_d, L_d]$ is C_d^* . Among them $k \in (0, 1)$.

$$C_d^* = k \cdot (U_d + L_d) - C_d \quad (14)$$

The integration of generalized opposition-based learning into the algorithmic framework can open up new search spaces at a time when population diversity is lost. Thus, populations involved in evolution are new populations composed of the best-performing individuals from the original population and the broadly opposing populations. A common strategy for engaging opposing populations in the evolutionary process in what form and in what state, is a decreasing probability, but this strategy generally loses its effectiveness in the later stages of convergence of populations. Therefore, we choose to generate the opposing population according to the random probability to ensure that the generalized opposition-based learning can play an exploration of the search space in the whole evolutionary process. The convergence of the algorithm should theoretically be accelerated in late evolution because the algorithm employs an elite retention strategy during execution, and therefore generalized opposition-based learning does not affect the convergence in the late evolution. Based on this, the execution pseudo-code for the proposed MGOEO is given in Algorithm 2. The important part is that lines 12 to 18 are the cross and mutation operations of the algorithm; lines 19 to 21 are the generation of generalized opposite popu-

lations; lines 22 to 24 are the selection of dominant individuals to form the new population.

4. Numerical experiments

4.1. Experimental design

Numerical experiments are designed in this sub-section to verify the performance of MGOEO, using the 23 functions provided in the literature (Faramarzi et al., 2020) as benchmarks for testing. The functions F1-F7 are unimodal, F8-F13 are high-dimensional multi-modal, F14-F23 are fixed-dimensional multi-modal. The comparison algorithm used is the original algorithm EO, FPA, SSA and SCA.

To avoid the impact of random initialization of populations on the experimental results, we perform 30 independent runs of all algorithms in all experiments. In addition, non-parametric tests are performed on the experimental results. The relevant parameter settings: the population number: $N=30$; maximum iterations number: $Maxgen=1000$, $a_1=2$ in EO, The same parameters in MGOEO and EO are consistent, in addition the probability $P=0.5$; the conversion probability in FPA is 0.2 (Draa, 2015). The parameter r_1 of SSA is $r_1 = 2 * \exp(-4 * t / Maxgen)^2$ and parameter r_1 of SCA is $r_1 = 2 - t * (2 / Maxgen)$.

4.2. Discussion of results

To test the performance of MGOEO, we use it to solve for the optimal values of the function F1-F23. Table 1 gives the mean value and standard deviation of the five algorithms in 30 independent runs. Fig. 2 gives representative convergence curves of these algorithms, from which the convergence speed and the ability to jump out of local limitation can be observed.

First of all, we can learn from the comparison of mean values and Wilcoxon rank-sum in Table 1. The optimization effect of MGOEO on F6 and F21 are weak compared with EO, and the optimization effect on F9, F14 and F16-F19 are consistent with EO, while there is no significant difference in Wilcoxon

Algorithm 2 : Algorithm MGOEO

Input: $Maxgen, NP, P$ **Output:** $Gbest$

```
1: Initialization population ;
2: while ( $t < Maxgen$ ) do
3:   for  $i=1:NP$  do
4:     Calculate the fitness of  $C_i$ ;
5:   end for
6:    $C = \text{sort}(C)$ 
7:    $C_1$ - $C_4$  and their arithmetic mean to the equalization pool;
8:   for  $i=1:NP$  do
9:     Execute the relevant equation to calculate the new population posi-
       tion, expressed as  $NewC$ ;
10:    Calculate the fitness of  $NewC_i$ ;
11:     $r_1 = \text{rand}$ ;
12:     $NewM\_C_i = C_i + r \cdot (C_j - C_k)$ 
13:    if  $t_1$  was no change for ten consecutive times then
14:       $N\_C_i = NewM\_C_i$ 
15:    else
16:       $N\_C_i = NewC_i$ 
17:    end if
18:  end for
19:  if  $rand < P$  then
20:    The new population  $OPOP$  was generated by using generalized po-
      sition based learning;
21:  end if
22:   $NPOP = N\_C \cup OPOP$ 
23:  Storing the  $NP$  elements of the set  $NPOP$  into  $N\_C$ 
24:  Compare  $C$  with  $N\_C$  and save the smaller value in  $C$ 
25:   $Gbest = \min(C)$ 
26: end while
```

rank-sum. The effect of MGOEO on F20-F22 is not as well as that of FPA. The
255 optimization effect of MGOEO is the same for F16-F19 and F23.

Compared to SSA, MGOEO obtains the same mean value for F16-F18, how-
ever, it not as good as SSA for F21. Compared to SCA, MGOEO obtains the
same objective solution for F16-F19, and MGOEO is not as accurate as SCA
for F21. In general, MGOEO is superior to EO, FPA, SSA and SCA in per-
260 formance of 15, 15, 18 and 18 functions respectively when solving the objective
solutions of 23 test functions. Therefore, we believe that MGOEO improves the
objective precision of solving function compared to EO, and the optimization
performance is better than the other three algorithms. Those 23 test functions
used can be divided into three categories, and we further performed Friedman
265 rank-sum tests according to their categories. Its comparative results are pre-
sented in Table 2. The result of Friedman rank-sum test shows that MGOEO is
the best on the whole functions, unimodal functions and the statistical results
of high-dimensional multimodal function. When dealing with fixed-dimension
problems, its performance is slightly inferior to FPA. This is also because FPA
270 can search for a better solution when facing F20-F22 function.

The evolutionary curves of the relevant algorithms are given in Fig.2, where
only representative figures are given. It can be seen from the F1 in Fig.2 that
the convergence speed and precision of MGOEO have obvious advantages. We
do not give the function evolution curve of F2-F4 here, because they are both
275 convergence cases similar to F1. The degree of final convergence of MGOEO in
optimizing F5 is essentially similar to that of FPA and SCA, but it can be seen
that MGOEO has a relatively better convergence speed. Observing the evolu-
tion of EO and MGOEO from F7, the two algorithm curves are intertwined,
with EO having higher solving precision than MGOEO in the medium term,
280 and MGOEO having stronger search ability in both pre- and post-evolutionary
stages. Both MGOEO and EO can search for the optimal solution when solving
the function F9, but MGOEO converges faster. The final convergence **effects** of
MGOEO and SCA are better than EO when processing F11, where MGOEO
can reach the theoretical optimal value. From the evolutionary curve of F14,

285 it was observed that the final convergence curve of MGOEO essentially over-
 lapped with EO, FPA, and the final convergence precision was consistent. We
 do not give the evolutionary graphs of the functions F16-F19 because they are
 similar to F14 and the convergence curves of the algorithm essentially overlap.
 According to the evolution curve of F15 in Fig.2, the optimization effect of FPA
 290 is weaker than that of EO before 200 iterations, and afterwards the precision
 exceeds the EO to fall into local optimization at 300 iterations. MGOEO is the
 best from the beginning in terms of both convergence speed and convergence
 precision when dealing with the F15 function. As can be seen from the F23
 in Fig.1, MGOEO and FPA have the best convergence precision, however, the
 295 convergence speed of MGOEO is quicker than that of FPA. We can see from the
 iterative curves of the functions F6 and F22 that the optimization of MGOEO is
 not ideal for both problems. It is not possible for anyone optimization algorithm
 to obtain satisfactory solutions for all optimization problems. MGOEO can
 capture solutions with higher precision for most function optimization, which
 300 show that MGOEO is a valid and feasible algorithm.

5. Multi-UAV flight path simulation

5.1. Describe the coding of multi-UAV flight path

The multi-UAV flight path planning problem requires a solution in which
 continuous space is divided into discrete points combined into trajectories. A
 305 UAV's trajectory is a collection of discrete points. In this paper, we use the
 scheme of semi-random neighbor selection ([Dewangan et al., 2019](#)) for coding.
 Each location node has six possible neighbors in three-dimensional space. The
 common strategy is to select one of the feasible nodes and then step by step
 planning. But, this method is to select three feasible neighbors for path mi-
 310 gration, or non-viable neighbors if the selected point is in an obstacle, which
 simplifies the path calculation length to the target point. In addition, the UAV's
 path needs to be as smooth as possible, avoiding sharp turns or large angular

Figure 2: Convergence curves for test functions

F11

F12

F14

F15

F22

F23

Figure 2: Continued

Table 1: Experimental results of functions and Wilcoxon rank-sum statistical comparison, reference MGOEO

ID	EO			FPA			SSA			SCA			MGOEO		
	mean	std		mean	std		mean	std		mean	std		mean	std	
F1	1.57E-86	6.71E-86	+	1.12E-08	3.01E-09	+	1.10E-08	2.79E-09	+	1.59E-36	1.46E-34	+	2.95E-194	0.00E+00	
F2	1.57E-48	3.95E-48	+	8.44E+17	4.59E+18	+	4.46E+19	1.09E+16	+	2.39E-26	5.95E-26	+	1.25E-100	5.05E-100	
F3	1.91E-17	1.04E-16	+	2.19E+02	1.48E+02	+	2.38E+02	3.24E+02	+	4.98E+01	3.11E+01	+	3.24E-139	1.01E-138	
F4	3.81E-19	1.83E-18	+	5.41E+00	2.81E+00	+	5.27E+00	3.21E+00	+	3.38E+00	9.76E+00	+	3.80E-86	1.85E-85	
F5	2.44E+01	1.52E-01	+	2.09E+02	4.42E+02	+	1.06E+02	2.29E+02	+	2.69E+01	3.16E-01	+	2.25E+01	5.21E-01	
F6	2.01E-09	3.59E-09	-	1.14E-08	3.36E-09	+	1.17E-08	2.63E-09	+	8.06E-02	1.27E-01	+	7.03E-09	1.11E-08	
F7	3.71E-04	2.81E-04	+	7.93E-02	3.13E-02	+	8.73E-02	3.47E-02	+	4.01E-03	3.34E-03	+	1.54E-04	9.37E-05	
F8	-8.97E+03	6.97E+02	+	-7.56E+03	9.48E+02	+	-7.28E+03	7.22E+02	+	-5.50E+03	4.39E+02	+	-9.03E+03	7.85E+02	
F9	0.00E+00	0.00E+00	=	7.10E+01	2.35E+01	+	7.08E+01	2.55E+01	+	2.03E-10	1.53E-12	+	0.00E+00	0.00E+00	
F10	4.68E-15	9.01E-16	+	1.31E+00	8.13E-01	+	1.52E+00	7.36E-01	+	1.01E+01	1.00E+01	+	8.88E-16	0.00E+00	
F11	4.11E-04	2.25E-03	+	1.08E-02	1.13E-02	+	6.48 E-03	1.16E-02	+	3.85E-10	7.36E-08	+	0.00E+00	0.00E+00	
F12	6.91E-03	2.63E-02	+	3.82E+00	2.48E+00	+	2.44E+00	1.66E+00	+	1.30E-02	7.13E-03	+	7.12E-10	1.92E-09	
F13	2.34E-02	3.87E-02	+	1.65E+00	4.61E+00	+	2.19E+00	7.34E+00	+	7.04E-01	2.67E-01	+	2.00E-02	4.09E-02	
F14	9.98E-01	3.62E-01	=	9.98E-01	7.66E-01	+	1.09E+00	3.03E-01	+	1.06E+01	4.31E-01	+	9.98E-01	3.62E-01	
F15	4.38E-03	8.13E-03	+	3.99E-04	5.93E-03	+	3.48E-03	3.01E-04	+	6.77E-04	1.78E-04	+	3.68E-04	5.10E-03	
F16	-1.03E+00	6.58E-16	=	-1.03E+00	5.53E-14	=	-1.03E+00	2.24E-14	=	-1.03E+00	1.05E-07	=	-1.03E+00	6.52E-16	
F17	3.98E-01	0.00E+00	=	3.98E-01	6.17E-14	=	3.98E-01	9.78E-14	=	3.98E-01	8.06E-06	=	3.98E-01	0.00E+00	
F18	3.00E+00	6.55E-16	=	3.00E+00	2.01E-13	=	3.00E+00	3.38E-13	=	3.00E+00	1.58E-06	=	3.00E+00	1.20E-15	
F19	-3.86E+00	1.44E-03	=	-3.84E+00	1.41E-01	=	-3.84E+00	2.77E-08	+	-3.86E+00	1.90E-03	=	-3.86E+00	2.70E-15	
F20	-3.25E+00	6.68E-02	+	-3.28E+00	6.32E-02	-	-3.25E+00	5.55E-02	+	-3.24E+00	7.60E-02	+	-3.27E+00	6.30E-02	
F21	-8.62E+00	2.06E+00	-	-1.01E+01	2.93E+00	-	-8.13E+00	2.69E+00	-	-8.75E+00	2.51E+00	-	-7.77E+00	2.59E+00	
F22	-9.59E+00	2.18E+00	+	-1.04E+01	2.96E+00	-	-8.19E+00	2.90E+00	+	-8.93E+00	1.61E+00	+	-9.69E+00	2.16E+00	
F23	-9.64E+00	2.39E+00	+	-1.05E+01	3.34E+00	=	-8.84E+00	3.12E+00	+	-1.03E+01	9.82E-01	+	-1.05E+01	2.29E-15	
R+ / R-				15/2/6			15/3/5			19/1/3			18/1/4		

turns, and the smoothing of the UAV's path is handled using Polynomial of degree five (Dewangan et al., 2019; Jain et al., 2019; Saxena et al., 2019).

5.2. Experimental design instructions

In the previous section, we tested and validated the continuous optimization capabilities of MGOEO. In this subsection we use the proposed MGOEO to solve the problem of multi-UAVs flight path planning. The three-dimensional geography of the multi-UAVs flight uses the map coordinates provided in the literature (Dewangan et al., 2019; Saxena et al., 2019). Five UAVs are used for flight path optimization, Table 3 gives the start and goal coordinates of the UAVs in three map environments, and Table 4 gives the obstacle location coordinates in three map environments. In this paper, we experimentally verify the performance of the MGOEO in handling the multi-UAVs flight path planning problem, in addition to comparing it with the EO algorithm and comparing it with eight other algorithms such as PSO, IBA, SCA, GWO, SSA, etc., the specific setup parameters of these eight algorithms are described in literature

Table 2: Friedman rank-sum statistical comparison by test function classification

General rank	F1-F7		F8-F13		F14-F23		All Functions	
	Algorithm	Rank mean	Algorithm	Rank mean	Algorithm	Rank mean	Algorithm	Rank mean
1	MGOEO	1.14	MGOEO	1.1	FPA	2.2	MGOEO	1.73
2	EO	1.86	EO	1.9	MGOEO	2.45	EO	2.45
3	SCA	3.29	SCA	3.8	EO	3.15	FPA	3.27
4	FPA	4.29	FPA	4.0	SCA	3.3	SCA	3.41
5	SSA	4.43	SSA	4.2	SSA	3.9	SSA	4.14

(Dewangan et al., 2019; Saxena et al., 2019). All the algorithms used in the experiment have a population size of 25 and a maximum number of iterations of 35. The experiments here are simulated on Matlab2014A.

Table 3: Start and Goal position of 3D map

Map	Start Position					Goal Position				
Map 1	(2, 10, 2)	(1, -4, 1)	(9.2, 17, 3)	(9.2, 10, 3)	(0.1, 10, 2)	(1, -4, 1)	(0.1, 17, 3)	(9, -4, 1)	(0.9, -4, 5)	(9, 10, 2)
Map 2	(0, 1, 5)	(0, 2, 5)	(0, 3, 5)	(19, 4, 5)	(19, 5, 5)	(19, 0, 5)	(19, 5, 5)	(19, 4, 5)	(0, 3, 5)	(0, 1, 5)
Map 3	(2, 10, 2)	(1, -4, 1)	(9.2, 17, 3)	(9.2, 10, 3)	(0.1, 10, 2)	(1, -4, 1)	(0.1, 17, 3)	(9, -4, 1)	(0.9, -4, 5)	(9, 10, 2)

330

5.3. Discuss the experimental results

Simulation experiments are conducted for multi-UAVs flight path planning in three geographic environments. Fig.3 is five UAVs path map which is optimized based on MGOEO in the Map1 environment. Five paths can be observed that do not affect each other, successfully avoiding obstacles while having a relatively smooth path. Fig.4 is the iterative curve of MEOGO, EO and other eight optimization algorithms used to solve the overall objective of multi-UAVs flight plan planning in Map1 environment. We can learn from Fig.4 that during the initial iteration, MGOEO performs worse than algorithms such as EO. The generalized opposition-based learning probability we set allows MGOEO to learn opposing populations in the pre-evolutionary period, so a larger coverage of the search space is naturally less efficient, but the MGOEO's objective solution

340

Table 4: Location coordinates of obstacles in 3D map

Obstacle	Map1	Map2	Map3
N1	(0, 2, 0) - (10, 2.5, 1.5)	(3.1, 0, 2.1) - (3.9, 5, 6)	(0, -2, 0) - (10, -1.5, 1.5)
N2	(0, 2, 4.5) - (10, 2.5, 6)	(9.1, 0, 2.1) - (9.9, 5, 6)	(0, 2, 0) - (10, 2.5, 1.5)
N3	(0, 2, 1.5) - (3, 2.5, 4.5)	(15.1, 0, 2.1) - (15.9, 5, 6)	(0, 2, 4.5) - (10, 2.5, 6)
N4	(7, 2, 1.5) - (10, 2.5, 4.5)	(0.1, 0, 0) - (0.9, 5, 3.9)	(0, 2, 1.5) - (3, 2.5, 4.5)
N5	(0, 15, 0) - (10, 20, 1)	(6.1, 0, 0) - (6.9, 5, 3.9)	(0,-2, 3) - (10, -1.5, 5.5)
N6	(0, 15, 1) - (10, 16, 3.5)	(12.1, 0, 0) - (12.9, 5, 3.9)	(3, 0, 2.4) - (7, 0.5, 4.5)
N7	(0, 18, 4.5) - (10, 19, 6)	(18.1, 0, 0) - (18.9, 5, 3.9)	(7, 2, 1.5) - (10, 2.5, 4.5)
N8	(3, 0, 2.4) - (7, 0.5, 4.5)	NULL	(0, 15, 0) - (10, 20, 1)
N9	NULL	NULL	(0, 15, 1) - (10, 16, 3.5)
N10	NULL	NULL	(0, 18, 4.5) - (10, 19, 6)
N11	NULL	NULL	(0, 7, 0) - (10, 7.5, 0.5)
N12	NULL	NULL	(0, 7, 2) - (10, 7.5, 5.5)
N13	NULL	NULL	(0, 11, 0) - (10, 11.5, 2.5)
N14	NULL	NULL	(0, 11, 4) - (10, 11.5, 5.5)
N15	NULL	NULL	(0, -2, 1.5) - (3, -1.5, 3)
N16	NULL	NULL	(6, -2, 1.5) - (10, -1.5, 3)

Fig.3 Multi-UAVs path under the Map1
Based on MEOGO

Fig.4 Iterative curves of ten algorithms under the Map1

Fig.5 Multi-UAVs path under the Map2
Based on MEOGO

Fig.6 Iterative curves of ten algorithms under the Map2

Fig.7 Multi-UAVs path under the Map3
Based on MEOGO

Fig.8 Iterative curves of ten algorithms under the Map3

precision exceeds EO after the 5th iteration. Ultimately, the MGOEO's search solution is the best of the ten algorithms after 20th iterations.

345 Fig.5 is the simulation map of UAVs flight path obtained by MGOEO in Map2 environment, the spatial density of obstacles in this environment is relatively large, we can still observe that the flight curve can avoid obstacles and the flight direction is smoothly towards the goal position. Fig.6 is a comparison of the iterative curves of the ten algorithms in a Map2 environment, where
 350 obstacle coverage is dense and where it is easy to search for unfeasible locations in a way point search. From Fig.6, it can be seen that MGOEO shows better performance than the other 9 algorithms at the beginning of the iteration, and eventually MGOEO can capture better positions and search for better objective solutions.

355 The Map3 environment is the one with the most obstacles of the three map environments used in this paper. Fig.7 is flight path schematic of MGOEO's handling of five UAVs in this environment, and we can observe that the UAV's flight curve can effectively avoid obstacles to reach the goal location safely. From Fig.8, we can observe the iterative curve of the cost of optimizing the overall
 360 path of five UAVs in the Map3 environment with ten algorithms, we can see that the algorithm MGOEO is only better than the algorithm IBA, BBO, SSA before 15 iterations, and after that, MGOEO is the best of the ten algorithms by jumping out of the local limit.

It is concluded that the optimization results of MGOEO are the best among
 365 the three map environments, and it can be considered that MGOEO is an algorithm that can effectively and efficiently solve the problem of multi-UAVs flight path planning.

The running time of an algorithm is another important measure of performance. Table 5 shows the running times of several algorithms in the three map
 370 environments. The running times are counted in seconds for each algorithm in the different map environments. Taking Map1 as an example, we can get the algorithms in order of time as $SSA < PSO < EO < BBO < SCA < MGOEO < GWO < GSO < WOA < IBA$, where SSA takes the shortest time. In general,

there is no great difference in the time spent by these algorithms. The algo-
375 rithm MGOEO proposed in this paper adds some running time compared to
its original version EO, and these time expenditures come from the opposition
population construction. Analysis from the perspective of asymptotic compu-
tational complexity. Define F to denote the evaluation objective function. The
computational complexity of EO can be expressed as $\mathcal{O}=(Maxgen \cdot NP \cdot F)$.
380 The computational complexity of MGOEO for the opposing population in terms
of probability P is $\mathcal{O}=(P \cdot NP \cdot F)$. The computational complexity of MGOEO
under the number of iterations $Maxgen$ is $\mathcal{O}=(Maxgen \cdot (NP \cdot F(1 + P)))$.
The frequency of $(Maxgen \cdot NP \cdot F \cdot P)$ is smaller than $(Maxgen \cdot NP \cdot F)$, so
the asymptotic time complexity of MGOEO is $\mathcal{O}=(Maxgen \cdot NP \cdot F)$ which is
385 the same as EO. Therefore, the execution time growth rate of MGOEO does not
increase as the problem size increases. Combined with the running time statis-
tics of these algorithms in Table 5, the running time of MGOEO is acceptable.

Table 5: The running time statistics of the algorithms

Algorithm	Map1(seconds)	Map2(seconds)	Map3(seconds)
GWO	60.7863	77.6254	66.7853
GSO	63.9688	99.4531	79.5000
PSO	43.9062	59.4844	54.7656
BBO	56.4688	70.2969	64.5469
IBA	68.1975	105.0673	92.0289
WOA	66.7346	101.9000	81.2928
SCA	56.7346	77.3892	65.2891
SSA	43.0943	58.4322	51.2839
EO	53.8208	68.7803	64.3372
MGOEO	60.4731	80.0812	74.6955

6. Conclusion and Prospect

390 In this paper, an equilibrium optimizer with generalized opposition-based learning for solving the problem of multi-UAVs path planning. The first step is to improve the solving precision of the algorithm to handle high-dimensional problems. We use cross-mutation operations and generalized opposition-based learning to improve the implementation framework of the algorithm, which can
395 increase population diversity and thus extend the global search capability of the algorithm. Subsequently, we performed numerical experiments with 23 test functions. The results of optimization and non-parametric statistics verify that MGOEO's overall continuous optimization ability is better than EO and other three comparison algorithms. Finally, the proposed MGOEO is applied to
400 multi-UAVs flight path planning. The five UAVs are uniformly used for track simulation in a map environment with three different obstacle settings.

The simulation results show that the MGOEO-based planning UAVs can get a safer and smoother path at the start and goal positions. The iterative curve of the overall objective optimization for multi-UAVs flight path planning reveals
405 that MGOEO shows better convergence and smaller objective solution in all three map environments compared to the nine algorithms: EO, GWO, GSO, PSO, BBO, IBA, WOA, SCA and SSA. As a result, MGOEO has greater optimization capabilities and can better handle multi-UAVs flight path planning.

The EO algorithm has just been proposed, and its learning strategy of using
410 equilibrium pool needs to be further developed. Our future work can be divided into three categories. Firstly, we analyze the influence of numerical parameters on the global and local search ability of EO algorithm, and analyze its convergence performance theoretically. Secondly, a new hybrid optimization strategy need be designed to improve the performance of the algorithm and expand
415 its application field. Thirdly, the environment of multi-UAVs flight path planning is static, so the algorithm will be designed for UAVs path planning in the dynamic environment of obstacles in the future.

Acknowledgments

This work was supported in part by the Postgraduate Research & Practice Innovation Program of Jiangsu Province(KYCX21_0226), China Scholarship Council Public Graduate Program at High Level Universities (202006830128)

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest with any person(s) or Organization(s).

Ethical standard This article does not contain any studies with human participants performed by any of the authors

References

- Abd Elaziz, M., Oliva, D., Ewees, A. A., & Xiong, S. (2019). Multi-level thresholding-based grey scale image segmentation using multi-objective multi-verse optimizer. *Expert Systems with Applications*, 125, 112–129.
- Abd Elaziz, M., Oliva, D., & Xiong, S. (2017). An improved opposition-based sine cosine algorithm for global optimization. *Expert Systems with Applications*, 90, 484–500.
- Abdel-Basset, M., Chang, V., & Mohamed, R. (2020). A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems. *Neural Computing and Applications*, (pp. 1–34).
- Cai, Y., Xi, Q., Xing, X., Gui, H., & Liu, Q. (2019). Path planning for uav tracking target based on improved a-star algorithm. In *2019 1st International Conference on Industrial Artificial Intelligence (IAI)* (pp. 1–6). IEEE.
- Chen, H., Wang, M., & Zhao, X. (2020). A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems. *Applied Mathematics and Computation*, 369, 124872.

- 445 Chen, H., Xu, Y., Wang, M., & Zhao, X. (2019a). A balanced whale optimization algorithm for constrained engineering design problems. *Applied Mathematical Modelling*, 71, 45–59.
- Chen, Y., & Pi, D. (2019). Novel fruit fly algorithm for global optimisation and its application to short-term wind forecasting. *Connection Science*, 31, 244–266.
- 450 Chen, Y., & Pi, D. (2020). An innovative flower pollination algorithm for continuous optimization problem. *Applied Mathematical Modelling*, 83, 237–265.
- Chen, Y., Pi, D., & Wang, B. (2019b). Enhanced global flower pollination algorithm for parameter identification of chaotic and hyper-chaotic system. 455 *Nonlinear Dynamics*, 97, 1343–1358.
- Dewangan, R. K., Shukla, A., & Godfrey, W. W. (2019). Three dimensional path planning using grey wolf optimizer for uavs. *Applied Intelligence*, 49, 2201–2217.
- Draa, A. (2015). On the performances of the flower pollination algorithm— 460 qualitative and quantitative analyses. *Applied Soft Computing*, 34, 349–371.
- Faramarzi, A., Heidarinejad, M., Stephens, B., & Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems*, 191, 105190.
- Guo, J., Liu, L., Liu, Q., & Qu, Y. (2009). An improvement of d* algorithm 465 for mobile robot path planning in partial unknown environment. In *2009 Second International Conference on Intelligent Computation Technology and Automation* (pp. 394–397). IEEE volume 3.
- Gupta, S., & Deep, K. (2019). A hybrid self-adaptive sine cosine algorithm with opposition based learning. *Expert Systems with Applications*, 119, 210–230.
- 470 Gururji, A. K., Agarwal, H., & Parsediya, D. (2016). Time-efficient a* algorithm for robot path planning. *Procedia Technology*, 23, 144–149.

- Jain, G., Yadav, G., Prakash, D., Shukla, A., & Tiwari, R. (2019). Mvo-based path planning scheme with coordination of uavs in 3-d environment. *Journal of Computational Science*, 37, 101016.
- 475 Lin, N., Tang, J., Li, X., & Zhao, L. (2019). A novel improved bat algorithm in uav path planning. *Computers, Materials & Continua*, 61, 323–344.
- Liu, J., Wu, J., & Liu, M. (2020). Uav monitoring and forecasting model in intelligent traffic oriented applications. *Computer Communications*, 153, 499–506.
- 480 Liu, Z., Li, Z., Zhu, P., & Chen, W. (2018). A parallel boundary search particle swarm optimization algorithm for constrained optimization problems. *Structural and Multidisciplinary Optimization*, 58, 1505–1522.
- Mahdavi, S., Rahnamayan, S., & Deb, K. (2018). Opposition based learning: A literature review. *Swarm and evolutionary computation*, 39, 1–23.
- 485 Qu, C., Gai, W., Zhang, J., & Zhong, M. (2020). A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (uav) path planning. *Knowledge-Based Systems*, (p. 105530).
- Radoglou-Grammatikis, P., Sarigiannidis, P., Lagkas, T., & Moscholios, I. (2020). A compilation of uav applications for precision agriculture. *Computer Networks*, 172, 107148.
- 490 Salgotra, R., & Singh, U. (2017). Application of mutation operators to flower pollination algorithm. *Expert Systems with Applications*, 79, 112–129.
- Sapre, S., & Mini, S. (2019). Opposition-based moth flame optimization with cauchy mutation and evolutionary boundary constraint handling for global optimization. *Soft Computing*, 23, 6023–6041.
- 495 Sarıççek, İ., & Akkuş, Y. (2015). Unmanned aerial vehicle hub-location and routing for monitoring geographic borders. *Applied Mathematical Modelling*, 39, 3939–3953.

- Saxena, P., Gupta, R., Maheshwari, A., Kaushal, G., & Tiwari, R. (2019). Three
500 dimensional route planning for multiple unmanned aerial vehicles using salp
swarm algorithm. *arXiv preprint arXiv:1911.10519*, .
- Shao, W., Pi, D., & Shao, Z. (2019). A pareto-based estimation of distribution
algorithm for solving multiobjective distributed no-wait flow-shop scheduling
problem with sequence-dependent setup time. *IEEE Transactions on Au-*
505 *tomation Science and Engineering*, 16, 1344–1360.
- Shao, Z., Pi, D., & Shao, W. (2020). Hybrid enhanced discrete fruit fly optimiza-
tion algorithm for scheduling blocking flow-shop in distributed environment.
Expert Systems with Applications, 145, 113147.
- Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on*
510 *evolutionary computation*, 12, 702–713.
- Tizhoosh, H. R., & Ventresca, M. (2008). *Oppositional concepts in computa-*
tional intelligence volume 155. Springer.
- Wang, G.-G., Chu, H. E., & Mirjalili, S. (2016). Three-dimensional path plan-
ning for ucav using an improved bat algorithm. *Aerospace Science and Tech-*
515 *nology*, 49, 231–238.
- Wang, H., Lyu, W., Yao, P., Liang, X., & Liu, C. (2015). Three-dimensional
path planning for unmanned aerial vehicle based on interfered fluid dynamical
system. *Chinese Journal of Aeronautics*, 28, 229–239.
- Wang, H., Wu, Z., Rahnamayan, S., Liu, Y., & Ventresca, M. (2011). Enhanc-
520 ing particle swarm optimization using generalized opposition-based learning.
Information Sciences, 181, 4699–4714.
- Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for opti-
mization. *IEEE transactions on evolutionary computation*, 1, 67–82.
- Yao, P., & Wang, H. (2017). Dynamic adaptive ant lion optimizer applied to
525 route planning for unmanned aerial vehicle. *Soft Computing*, 21, 5475–5488.

Yao, W., Qi, N., Wan, N., & Liu, Y. (2019). An iterative strategy for task assignment and path planning of distributed multiple unmanned aerial vehicles. *Aerospace Science and Technology*, 86, 455–464.

530 Zhang, J., Yu, J., Qu, X., & Wu, Y. (2017). Path planning for carrier aircraft based on geometry and dijkstra’s algorithm. In *2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE)* (pp. 115–119). IEEE.