International Journal of Computational Intelligence Systems
Vol. **13(1)**, 2020, *pp.* 1315–1331
DOI: https://doi.org/10.2991/ijcis.d.200825.001; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Research Article

# Quantum Behavior-Based Enhanced Fruit Fly Optimization Algorithm with Application to UAV Path Planning

Xiangyin Zhang*,[1,3,4], Shuang Xia[1,2,4], Xiuzhi Li[1,2,4]

[1] *Faculty of Information Technology, Beijing University of Technology, Beijing 100124, PR China*
[2] *Engineering Research Center of Digital Community, Ministry of Education, Beijing 100124, PR China*
[3] *Beijing Laboratory for Urban Mass Transit, Beijing 100124, PR China*
[4] *Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, PR China*

**ARTICLE INFO**

*Article History*

Received 19 Jan 2020
Accepted 23 Aug 2020

*Keywords*

Fruit fly optimization algorithm
Continuous function optimization
Delta potential well
Quantum behavior
Path planning

**ABSTRACT**

As a newly developed simple and effective optimization technology, the fruit fly optimization algorithm (FOA) has been successfully applied in many fields. To accelerate the algorithm convergence and avoid the local optimum, the enhanced FOA based on quantum theory called QFOA is proposed in this paper. When establishing the quantum Delta potential well around the location of fruit fly swarm, QFOA introduces the quantum behavior-based searching mechanism into the original osphresis-based search procedure of FOA. In the process that fruit flies find and move toward the food source, fruit flies follow the wave function property of the Delta potential well rather than the Newtonian mechanics. Taking advantage of the probability and uncertainty of quantum theory, the proposed QFOA can effectively overcome the weakness in premature convergence and easy trapping into local optimum. Since there are two popular models of the basic FOA, this paper also develops two corresponding QFOAs. Experimental results on various benchmark functions show that both the two QFOA models has overall better performance compared with the basic FOA as well as other FOA variants and other well-known optimization algorithms. In addition, the proposed QFOAs are also applied to unmanned aerial vehicle (UAV) path planning problem in the three-dimensional environment, and comparative results about the obtained optimal flight path and population convergence process show the effectiveness of QFOAs.

## 1. INTRODUCTION

The complex continuous optimization problem, which is to find the optimal solution that fulfills a set of constraints and achieves the maximum or minimum value of the function [1], is a universal issue, in many mathematical and engineering fields, such as automatic control, decision-making, signal processing, etc. Many traditional optimization techniques, which often use the gradient strategy, are applied to these problems, such as the integer programming, gradient descent methods, linear programming, and dynamic programming [2]. They are usually based on the single initial individual solution and will achieve the same local solutions when starting from the same initial individual. When dealing with complex practical problems that generally have high dimension, or multiple constraints, or multimodal [3], these traditional optimization techniques easily lead to the high computational complexity and even fail to determine the optimal solutions. To solve the complex optimization problems, population-based intelligent optimization techniques, which usually search for the feasible solution in some random fashion with starting from the initial solution set, have emerged and show their advantages in terms of calculate precision and time complexity [4,5]. These intelligent optimization techniques are generally inspired by the biological social behavior, such as ant colony optimization (ACO) [6], particle swarm optimization (PSO) [7], artificial bee colony (ABC) optimization [8], firefly algorithm (FA) [9], etc., and the evolution process in nature, including differential evolution (DE) algorithm [4,10], genetic algorithm (GA) [11], etc., as well as the physical or chemical phenomenon, such as chemical reaction optimization (CRO) algorithm [12], gravitational search algorithm (GSA) [13], biogeography-based optimization (BBO) [14] algorithm, etc.

The fruit fly optimization algorithm (FOA) is another newly proposed population-based intelligent optimization technique [15], which is originally inspired by the food-finding behavior of fruit flies. FOA is not only very effective to solve complex optimization problems, but also more simple and easy to implement than other existing population-based algorithms [16]. Firstly, the procedure of FOA only takes several lines to code the core part of the algorithm in any programming language. Secondly, FOA has no other additional control parameters besides the maximum number of iteration and population size. Therefore, FOA has widely attracted the attention of many researchers from various fields, including path planning [16,17], flowshop rescheduling [18], multi-dimensional knapsack

[*] *Corresponding author. Email: zhangxgyn@foxmail.com*

problem [19], parameter identification [20], controller parameter tuning [21], etc.

FOA has simple operation and acceptable searching capability, however, it also has some shortcomings of the premature convergence and slow convergent speed during the later evolution process. To overcome these shortcomings as well as improve the search efficiency and global search capability, various improved FOA variants have been put forward by researchers [1, 16, 18-20,22-26]. The improved fruit fly optimization (IFFO) algorithm [1] introduces a new parameter to adaptively control the search scope around the fruit fly swarm location and a new solution generating method to enhance the accuracy and convergence rate. The phase angle-encoded FOA with mutation adaptation mechanism ($\theta$-MAFOA) [12] adopts the mutation adaptation mechanism to enhance the balance between the exploitation and exploration ability of the basic FOA, and besides, the phase angle-based-encoded strategy for fruit fly locations is also adopted to achieve the better convergence performance. In the chaotic-enhanced fruit fly optimization algorithm (CFOA) [20], the chaotic sequence is employed to enhance the global optimization capacity of the original FOA. In the multi-swarm fruit fly optimization algorithm (MFOA) [24], several sub-swarms move independently in the search space at the same time and local behavior rules between sub-swarms are employed to explore global optimal. All these improved FOA variants succeed in overcoming the drawback of the original FOA to some degree. However, for the FOA variants, there is still much room for improvement to achieve better performance in terms of either convergence speed or avoiding to trap into the local optimum, especially for the complex practical problems. Generally, the population diversity and the search randomness play the crucial roles in the population-based optimization techniques. In this paper, we proposed the quantum-behaved FOA (QFOA) model, with the purpose that try to take advantage of the uncertainty of quantum theory to further enhance the performance of FOA by enabling fruit flies to have the quantum behavior. Especially, for the two popular standard FOA models in the literatures, two QFOA models are both presented accordingly, as well as the implement processes.

The idea behind quantum theory, of which the theoretical basis is the classical quantum mechanics, is to hijack some of the “spookiness” in the area of physics known as quantum behavior mechanism [27]. For the population-based optimization techniques, solutions can usually show the higher diversity, which helps to achieve the powerful searching capability, when moving according to the quantum behavior rule. For example, the quantum-behaved particle swarm optimization (QPSO) makes particles move following quantum mechanics rather than Newtonian mechanics, and thus, help to enhance the particle’s ability of jumping out of the local optimum [28,29]. The quantum firefly algorithm (QFA) integrates quantum theory into the basic FA and succeed in overcoming the loss of diversity [30].

In this paper, we introduce the quantum theory into the original FOA to achieve more effective performance. In the enhanced FOA, we assume that the fruit flies locate in a quantum world and the Delta potential well exists around the location of fruit fly swarm. When moving in the space to search for the food, fruit flies follow the quantum mechanics and their behaviors have the uncertainty as the wave function. The quantum-behaved strategy can effectively avoid the premature convergence and improve the escaping ability from the local optimum. Numerical results on benchmark functions reveal that the proposed FOA models are superior to the standard FOA models and other FOA variants as well as some state-of-the-art population-based techniques. Then, we apply the proposed QFOA models to the practical complex problem that finds the optimal flight path for unmanned aerial vehicle (UAV) in the three-dimensional (3D) terrain environments with various enemy weapons. Numerical experiments are carried out on various test scenarios and the results show the effectivity of our proposed QFOA models.

The rest of this paper is organized as follows: Overview of the basic FOA is described in Section 2, including common descriptions of two FOA models. Section 3 presents detailed statement and implement process for the models of the proposed QFOA variants. In Section 4, the comparison and simulation results on several benchmark functions are studied. Section 5 presents the application of the proposed QFOA models to the UAV path planning. Experimental results of path planning are shown in Section 6. Section 7 concludes this paper finally.

## 2. STANDARD FOA MODELS

FOA is a new population-based algorithm for searching global optimization based on the foraging behavior of fruit flies [11]. Fruit flies are superior to other species in terms of olfactory and visual senses. By using their osphresis, fruit flies can find all types of scents floating in the air and explore the direction of food source. After that, fruit flies can find the food source when getting close to the location of food by using their vision.

According to the foraging characteristics of the fruit fly swarm, FOA can be divided into three consecutive operations, namely swarm location initialization, osphresis-based search, and vision-based search. As to the detailed implementation, existing literatures give two different types of mathematical models to describe the three operations, which are summarized and recorded as FOA-1 and FOA-2 in this paper. Both the two FOA models will be described in the following section:

Without loss of generality, the global optimization problem in the continuous domain can be summarized as follows:

$$\begin{aligned} &\min f(x) \\ &\text{s.t.} \quad x_j \in [X_{\min}, X_{\max}], j = 1, 2, ..., D \end{aligned} \tag{1}$$

where $f(\boldsymbol{x})$ is the objective function with the $D$-dimensional decision variable $\boldsymbol{x} = [x_1, x_2, ..., x_D]$, and $X_{\min}$ and $X_{\max}$ are the lower and upper bounds of the decision space, respectively.

### 2.1. FOA-1 Model

In 2012, W.T. Pan put forward the basic idea of FOA for the first time [15], and meanwhile, the complete mathematical model was given to describe the three operations of FOA. In this type of FOA model, the decision variable is encoded as the reciprocal of the distance between the fruit fly location and the origin. To avoid confusion, the

FOA model appearing in Refs. [9,15,17,19], is recorded as FOA-1, of which the operations are described as follows:

1. Swarm location initialization

Set the population size $M_{\text{osp}}$ and the maximum number of iterations $G_{\max}$, and then, the fruit fly swarm location $(X_{\text{axis}}, Y_{\text{axis}})$ is randomly initialized in the decision space as

$$\begin{cases} X_{\text{axis}} = rand \times (X_{\max} - X_{\min}) + X_{\min} \\ Y_{\text{axis}} = rand \times (Y_{\max} - Y_{\min}) + Y_{\min} \end{cases} \tag{2}$$

where *rand* is the uniform random number within [0,1].

2. Osphresis-based search

In the osphresis-based search process, the fruit fly swarm generate $M_{\text{osp}}$ new locations of the food source that are randomly around the current location of the fruit fly swarm. The random searching strategy using the osphresis to find the new food location is given by

$$\begin{cases} X_i = X_{\text{axis}} + random \\ Y_i = Y_{\text{axis}} + random \end{cases} \tag{3}$$

where $i = 1, 2, \ldots, M_{\text{osp}}$, and *random* is the random value in the range of [−1,1].

For every location generated by the osphresis-based searching strategy, its smell concentration judgment $S_i$ is the reciprocal of the distance between the location and the origin, as follows:

$$\begin{cases} Dist_i = \sqrt{X_i^2 + Y_i^2} \\ S_i = 1/Dist_i \end{cases} \tag{4}$$

Actually, the smell concentration judgment corresponds to the candidate solution of the objective function in the decision space. The value of the objective function for the candidate solution, which corresponds to the smell concentration of the food source for the smell concentration judgment, is calculated as follows:

$$Smell_i = f(S_i) \tag{5}$$

3. Vision-based search

The vision-based searching process of the fruit fly swarm is essentially a greedy selection procedure. The fruit fly swarm observe all the locations generated via the above osphresis-based searching process and select the best location with the minimum smell concentration, as follows:

$$index = \arg\min\,(Smell_i) \tag{6}$$

The best smell concentration $Smell_{index}$ is compared with $Smell_{\text{axis}}$ of the fruit fly swarm location $(X_{\text{axis}}, Y_{\text{axis}})$. If $Smell_{index} < Smell_{\text{axis}}$, then the fly swarm will fly toward the new location by using vision and $(X_{\text{axis}}, Y_{\text{axis}})$ will be replaced by $(X_{index}, Y_{index})$. The vision-based search can be described as follows:

$$\begin{cases} Smell_{\text{axis}} = Smell_{index} \\ X_{\text{axis}} = X_{index} \\ Y_{\text{axis}} = Y_{index} \end{cases} \tag{7}$$

The osphresis-based and vision-based searching processes are repeated until the stopping criterion is satisfied. The general procedure of abovementioned operations is shown in Algorithm 1.

**Algorithm 1:** Procedure of FOA-1

```
/* Initialization */
Set the generation counter $g = 0$;
Set the number of fruit flies as $M_{osp}$;
Randomly initialize the fruit fly swarm's location as $[X_{axis}, Y_{axis}]$;
/* Iterative search */
while termination criteria is not satisfied do
    Generation counter $g = g + 1$
    /* Search using osphresis */
    for $i = 1: M_{osp}$ do
        Generate the random direction and distance as:
          $X_i = X_{axis} + random,\ Y_i = Y_{axis} + random$
        Compute the distance to the origin as: $D_i = \sqrt{X_i^2 + Y_i^2}$
        Calculate judged value of smell concentration as: $S_i = 1/D_i$
        Evaluate the smell concentrations as: $Smell_i = f(S_i)$
    end for
    /* Search using vision */
    Select the fruit fly that has the best smell concentration:
        $index = \arg\max(Smell_i)$
    if $[X_{index}, Y_{index}]$ has the better smell than $[X_{axis}, Y_{axis}]$, then
        $X_{axis} = X_{index}$ and $Y_{axis} = Y_{index}$
end while
```

## 2.2. FOA-2 Model

In addition to the above FOA-1 model, there existing another popular mathematical model to describe operations of FOA [1,23], which is recorded as FOA-2 in this paper. The FOA-2 model codes the decision variable using the floating number directly and can be adapted to solve the high-dimensional functions [1]. The operations of FOA-2 model are described as follows:

1. Swarm location initialization

Firstly, the fruit fly swarm location $\boldsymbol{x}_{\text{axis}} = [x_{\text{axis},1}, x_{\text{axis},2}, \ldots, x_{\text{axis},D}]$ is randomly initialized as

$$x_{\text{axis},j} = rand \times (X_{\max} - X_{\min}) + X_{\min} \tag{8}$$

where *rand* is the uniform random generator number in the range of [0,1], $j = 1,2,\ldots,D$.

2. Osphresis-based search

In the osphresis-based search process, the $M_{\text{osp}}$ new locations are randomly generated around the current swarm location as

$$x_{ij} = x_{\text{axis},j} + random \tag{9}$$

where $i = 1, 2, \ldots, M_{\text{osp}}$, $j = 1,2,\ldots,D$, and *random* is the random value in the range of [−1,1].

Then, the smell concentration is calculated as follows:

$$Smell_i = f(\mathbf{x}_i) \tag{10}$$

where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, ..., x_{iD}]$.

3. Vision-based search

The vision-based searching process of the fruit fly swarm is essentially a greedy selection procedure. That is, the fruit fly swarm observer all the locations generated via the above osphresis-based searching process and find out the best location that has the minimum smell concentration, as follows:

$$index = \arg\min\left(Smell_i\right) \tag{11}$$

Compare the best smell concentration $Smell_{index}$ with the smell concentration $Smell_{\text{axis}}$ of the current swarm location $\boldsymbol{x}_{\text{axis}}$. If $Smell_{index} < Smell_{\text{axis}}$, then the fly swarm will leave $\boldsymbol{x}_{\text{axis}}$ and fly toward the new location $\boldsymbol{x}_{index}$ by using vision, as follows:

$$\begin{cases} Smell_{\text{axis}} = Smell_{index} \\ \quad \mathbf{x}_{\text{axis}} = \mathbf{x}_{index}, \end{cases} \quad \text{if} \quad Smell_{index} < Smell_{\text{axis}} \tag{12}$$

General procedure of the FOA-2 model is shown in Algorithm 2.

**Algorithm 2:** Procedure of FOA-2

```
/* Initialization */
Set the generation counter $g = 0$;
Set the number of fruit flies as $M_{\text{osp}}$;
Randomly initialize $\boldsymbol{x}_{\text{axis}} = [x_{\text{axis},1}, x_{\text{axis},2}, ..., x_{\text{axis},D}]$ with $[X_{\min}, X_{\max}]^D$;
/* Iterative search */
while termination criteria is not satisfied do
    Generation counter $g = g + 1$
    /* Search using osphresis */
    for $i = 1: M_{\text{osp}}$ do
        Generate random location as $x_{ij} = x_{\text{axis},j} + random, j = 1,2,...D$
        if $x_{ij} > X_{\max}$, then
            $x_{ij} = X_{\max}$
        if $x_{ij} < X_{\min}$, then
            $x_{ij} = X_{\min}$
        Evaluate smell concentrations $Smell_i = f(\boldsymbol{x}_i)$, $\boldsymbol{x}_i = [x_{i1}, x_{i2},..., x_{iD}]$
    end for
    /* Search using vision */
    Select the fruit fly that has the best smell concentration:
    $index = \arg\max(Smell_i)$
    if $f(\boldsymbol{x}_{index}) < f(\boldsymbol{x}_{\text{axis}})$, then
        $\boldsymbol{x}_{\text{axis}} = \boldsymbol{x}_{index}$
end while
```

## 3. QUANTUM-BEHAVED FOA MODELS

Based on the quantum theory, the fruit fly swarm are assigned to move in the quantum multi-dimensional space. The Delta potential well model is established to increase the uncertainty that the fruit fly sense and move to the food.

### 3.1. Quantum Delta Potential Well

According to the quantum theory [28], objects are described by the wave function $\psi(\boldsymbol{x}, t)$, rather than the position $\boldsymbol{x}$ and velocity $\boldsymbol{v}$. Any quantum object has the wave-like properties and can exist in many places at once. In the quantum space, the probability of the object appearing on a spot is proportional to the strength module of the wave function on this spot, as follows:

$$|\psi(x, t)|^2 \,\mathrm{d}x\mathrm{d}y\mathrm{d}z = Q\mathrm{d}x\mathrm{d}y\mathrm{d}z \tag{13}$$

where $Q\mathrm{d}x\mathrm{d}y\mathrm{d}z$ is the probability of the object appearing on the spot $\boldsymbol{x} = (x, y, z)$ at time $t$. Thus, the strength module $|\psi(\boldsymbol{x},t)|^2$ is the probability density function satisfying the following equation:

$$\int_{-\infty}^{+\infty} |\psi|^2 \,\mathrm{d}x\mathrm{d}y\mathrm{d}z = \int_{-\infty}^{+\infty} Q\mathrm{d}x\mathrm{d}y\mathrm{d}z = 1 \tag{14}$$

According to the quantum physics, the motion of objects is described by the Schrödinger's equation as follows:

$$i\hbar\frac{\partial}{\partial t}\psi(X, t) = \hat{H}\psi(X, t) \tag{15}$$

where $\hbar$ is a constant called Planck Constant, $\hat{H}$ is the Hamiltonian operator which is given by the following equation:

$$\hat{H} = -\frac{\hbar}{2m}\nabla^2 + V(x) \tag{16}$$

where $m$ is the object mass and $V(\boldsymbol{x})$ denotes the potential field of the object.

### 3.2. Quantum Behavior of Fruit Flies

Hypothesize the fruit fly swarm move in the quantum space, and fruit flies search for the food source in the Delta potential well. In the quantum space, the foraging and random search operations of fruit flies are replaced by the quantum behavior. Both the osphresis and vision of fruit flies become more uncertainty and random, which can help to increase the population diversity.

For simplicity, the one-dimensional space is considered here. If the location of the food source is denoted by $x$, its potential energy in the one-dimensional Delta potential well is represented as follows:

$$V(x) = -\gamma\delta(x - p_{\text{axis}}) = -\gamma\delta(y) \tag{17}$$

where the location of fruit fly swarm $p_{\text{axis}}$ is center of the Delta potential well.

According to the Schrödinger's equation [7,28], the following normalized wave function can be obtained as

$$\psi(y) = \frac{1}{\sqrt{L}}e^{-|y|/L} \tag{18}$$

where $L$ is the characteristic length of Delta potential well. Therefore, the probability density function is as follows:

$$Q(y) = |\psi(y)|^2 = \frac{1}{L}e^{-2|y|/L} \tag{19}$$

The above equation is equated to

$$y = \pm\frac{L}{2}\ln\frac{1}{u} \tag{20}$$

where $u$ is the random number uniformly distributed on (0, 1). Hence, we can obtain the food source location that the fruit fly searches for, which is determined as follows:

$$x = p_{\text{axis}} \pm \frac{L}{2}\ln\frac{1}{u} \tag{21}$$

## 3.3. Quantum-Behaved Searching Mechanism

In my proposed QFOA model, it assumes that a 1-D Delta potential well exists on each dimension at the swarm center attractor point, and the osphresis-based search operation has the quantum properties. The quantum-behaved foraging process of the fruit fly is depicted by the wave function instead of the completely random way.

Compared with the two standard FOA models, both two proposed QFOA models also contain three basic operations, including the swarm location initialization, the osphresis-based search using quantum-behaved mechanism, and the vision-based search. Unlike the standard FOA, the proposed QFOA models execute the quantum-behaved searching mechanism instead of the completely random osphresis-based search which are described in details as follows:

### 3.3.1. Quantum-behaved searching mechanism of QFOA-1

In the osphresis-based search process, $M_{\text{osp}}$ new locations of the food source are generated in the Delta potential well of which the swarm location ($X_{\text{axis}}$, $Y_{\text{axis}}$) is the center. Different from (9), the quantum-behaved searching mechanism for FOA-1 can be given by

$$\begin{cases} X_i = X_{axis} \pm \dfrac{L_{x,i}}{2} \ln \dfrac{1}{r_x} \\ Y_i = Y_{axis} \pm \dfrac{L_{y,i}}{2} \ln \dfrac{1}{r_y} \end{cases} \tag{22}$$

where $i = 1, 2, ..., M_{\text{osp}}$, $r_x$ and $r_y$ are the random value in the range of [0,1]. $L_{x,i}$ and $L_{y,i}$ are the Delta potential well characteristic length of the corresponding dimension, which are determined by the last searching location of the fruit fly based on the olfactory as follows:

$$\begin{cases} L_{x,i} = 2b\,|X_{axis} - X_i| \\ L_{y,i} = 2b\,|Y_{axis} - Y_i| \end{cases} \tag{23}$$

where $g$ is the current number of iteration and $b$ is contraction-expansion coefficient to control the quantum searching range, which is determined by

$$b = b_1 logsig\left(10 \cdot \left(0.5 - g/G_{\max}\right)\right) + b_2 \tag{24}$$

where $b_1$ and $b_2$ are the two parameters to restrict the value range, namely $b \in [b_2, b_1 + b_2]$.

### 3.3.2. Quantum-behaved searching mechanism of QFOA-2

As to the QFOA-2 model, fruit flies search the location of food source in the 1-D Delta potential well around the fruit fly swarm $\boldsymbol{x}_{\text{axis}} = [x_{\text{axis},1}, x_{\text{axis},2}, ..., x_{\text{axis},D}]$ and generate the new food location on each dimension by the following equation:

$$x_{ij} = x_{axis,j} \pm \frac{L_{ij}}{2} \ln \frac{1}{r} \tag{25}$$

where $i = 1, 2, ..., M_{\text{osp}}$, $j = 1, 2, ..., D$, and $r$ is a random number that uniformly distributed in [0,1]. The characteristic length $L$ is determined by

$$L_{ij} = 2b\left|x_{axis,j} - x_{ij}\right| \tag{26}$$

where the parameter $b$ is determined by (24).

### 3.3.3. Detailed procedure of QFOA models

The detailed implementation procedure of two QFOA models are presented in Algorithms 3 and 4, respectively.

**Algorithm 3:** Procedure of QFOA-1

```
/* Initialization */
Set the generation counter $g = 0$;
Set the number of fruit flies as $M_{\text{osp}}$; set the parameters $b_1$ and $b_2$;
Randomly initialize the fruit fly swarm's location as $[X_{axis}, Y_{axis}]$ and
ensure the judged value of smell concentration $S$ is within the search range
$[X_{\min}, X_{\max}]$;
/* Iterative search */
while termination criteria is not satisfied do
    Generation counter $g = g + 1$;
    Calculate parameter $b = b_1$ logsig(10 (0.5 −$g/G_{\max}$)) + $b_2$;
    /* Quantum-behaved search using osphresis */
    for $i = 1$: $M_{\text{osp}}$ do
        Calculate the characteristic length
            $L_{x,i} = 2b|X_{\text{axis}} - X_i|$, $L_{y,i} = 2b|Y_{\text{axis}} - Y_i|$;
        while 1 do
            Generate the random direction and distance as follows:
            if random > 0.5 then
                $X_i = X_{\text{axis}} + 0.5\ L_{x,j}\ \ln(1/r_x)$;
            else
                $X_i = X_{\text{axis}} - 0.5\ L_{x,j}\ \ln(1/r_x)$;
            if random > 0.5 then
                $Y_i = Y_{\text{axis}} + 0.5\ L_{y,j}\ \ln(1/r_y)$;
            else
                $Y_i = Y_{\text{axis}} - 0.5\ L_{y,j}\ \ln(1/r_y)$;
            Calculate the distance to the origin:
                $D_i = \sqrt{X_i^2 + Y_i^2}$;
            Calculate judged value of smell concentration:
                $S_i = 1/D_i$ ;
            if $S_i \le X_{\max}$ and $S_i \ge X_{\min}$ then
                break;
        end while
        Evaluate the smell concentrations:
            $Smell_i = f(S_i)$;
    end for
    /* Search using vision */
    Select the fruit fly that has the best smell concentration:
        $index = \arg\max(Smell_i)$;
    if $[X_{index}, Y_{index}]$ has the better smell than $[X_{axis}, Y_{axis}]$, then
        $X_{\text{axis}} = X_{index}$ and $Y_{\text{axis}} = Y_{index}$;
end while
```

## 4. RESULTS FOR FUNCTION OPTIMIZATION

To show the advantage of the proposed QFOA models, the standard FOA models [1,15], as well as some existing improved FOA

**Algorithm 4:** Procedure of QFOA-2

```
/* Initialization */
Set the generation counter $g = 0$;
Set the number of fruit flies as $M_{\text{osp}}$; set parameters $b_1$ and $b_2$;
Randomly initialize $\boldsymbol{x}_{\text{axis}} = [x_{\text{axis},1}, x_{\text{axis},2}, \ldots, x_{\text{axis},D}]$ with
  $[X_{\min}, X_{\max}]^D$;
/* Iterative search */
while termination criteria is not satisfied do
    Generation counter $g = g + 1$;
    Calculate the parameter $b = b_1$ logsig(10 (0.5 − g/$G_{\max}$)) + $b_2$;
    /* Quantum-behaved search using osphresis */
    for $i = 1: M_{\text{osp}}$ do
        for $j = 1: D$ do
            Calculate the characteristic length
              $L_{ij} = 2b|x_{\text{axis},j} - x_{ij}|$;
            while 1 do
                Generate the random location as follows:
                if random > 0.5 then
                    $x_{ij} = x_{\text{axis},j} + 0.5\, L_{ij} \ln(1/r)$;
                else
                    $x_{ij} = x_{\text{axis},j} - 0.5\, L_{ij} \ln(1/r)$;
                if $x_{ij} \le X_{\max}$ and $x_{ij} \ge X_{\min}$ then
                    break;
            end while
        end for
        Evaluate the smell concentrations:
          $Smell_i = f(\boldsymbol{x}_i)$, $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]$;
    end for
    /* Search using vision */
    Select the fruit fly that has the best smell concentration:
      $index = \arg\max(Smell_i)$;
    if $f(\boldsymbol{x}_{index}) < f(\boldsymbol{x}_{\text{axis}})$, then
        $\boldsymbol{x}_{\text{axis}} = \boldsymbol{x}_{index}$;
end while
```

variants, e.g., IFFO [1], MFOA [24], and $\theta$-MAFOA [16] are used to compare. Besides, three typical papulation-based algorithms are also compared with the proposed QFOA, including PSO [7], DE [4], ABC [8]. All algorithms are coded using Matlab-2009a according to their original references, and no commercial algorithm tools are used. The experiments are performed on a computer with Intel(R) Core(TM) i5-4590 @ 3.3GHz CPU and 8 GB RAM under Win-7 platform. Each algorithm is run 50 times independently for every benchmark function and the statistical results are used to evaluate and compare the performance of the algorithm.

## 4.1. Test Functions

We considered eight typical benchmark functions with dimensions equal to 10 and 20 (i.e., $D = 10$ and $D = 20$), which are commonly used in the literatures [1,7,24]. The test functions include four unimodal ($F_1$, $F_2$, $F_3$, $F_4$) and four multimodal functions ($F_5$, $F_6$, $F_7$, $F_8$). For all the tested functions, we set the optimum $\boldsymbol{o} = \mathbf{1}^D$.

1. $F_1$: Shifted Sphere Function

$$F_1(\boldsymbol{x}) = \sum_{i=1}^{D} z_i^2, \quad \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o},$$

$\boldsymbol{x} = [x_1, x_2, \cdots, x_D], o = [o_1, o_2, \cdots, o_D]$
where $\boldsymbol{x} \in [-100, 100]^D$, the global optimum $\boldsymbol{x}^* = \boldsymbol{o}$, and $F_1(\boldsymbol{x}^*) = 0$.

2. $F_2$: Shifted Schwefel's Problem 1.2

$$F_2(\boldsymbol{x}) = \sum_{i=1}^{D} \left( \sum_{j=1}^{i} z_j \right)^2, \quad \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o},$$

$\boldsymbol{x} = [x_1, x_2, \cdots, x_D], o = [o_1, o_2, \cdots, o_D]$
where $\boldsymbol{x} \in [-100, 100]^D$, the global optimum $\boldsymbol{x}^* = \boldsymbol{o}$, and $F_2(\boldsymbol{x}^*) = 0$.

3. $F_3$: Shifted Rotated High Conditioned Elliptic Function

$$F_3(\boldsymbol{x}) = \sum_{i=1}^{D} \left(10^6\right)^{\frac{i-1}{D-1}} z_i^2, \quad \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o},$$

$\boldsymbol{x} = [x_1, x_2, \cdots, x_D], o = [o_1, o_2, \cdots, o_D]$
where $\boldsymbol{x} \in [-100, 100]^D$, the global optimum $\boldsymbol{x}^* = \boldsymbol{o}$, and $F_3(\boldsymbol{x}^*) = 0$.

4. $F_4$: Shifted Step Function

$$F_4(\boldsymbol{x}) = \sum_{i=1}^{D} \left(\lfloor z_i + 0.5 \rfloor\right)^2, \quad \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o},$$

$\boldsymbol{x} = [x_1, x_2, \cdots, x_D], o = [o_1, o_2, \cdots, o_D]$
where $\boldsymbol{x} \in [-100, 100]^D$, the global optimum $\boldsymbol{x}^* = \boldsymbol{o}$, and $F_4(\boldsymbol{x}^*) = 0$.

5. $F_5$: Shifted Rosenbrock's Function

$$F_5(\boldsymbol{x}) = \sum_{i=1}^{D-1} \left[ 100 \left( z_i^2 - z_{i+1} \right)^2 + (z_i - 1)^2 \right], \quad \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o},$$

$\boldsymbol{x} = [x_1, x_2, \cdots, x_D], o = [o_1, o_2, \cdots, o_D]$
where $\boldsymbol{x} \in [-10, 10]^D$, the global optimum $\boldsymbol{x}^* = \boldsymbol{o}$, and $F_5(\boldsymbol{x}^*) = 0$.

6. $F_6$: Shifted Rotated Ackley's Function with Global Optimum on Bounds

$$F_6(\boldsymbol{x}) = \begin{matrix} -20 \exp\left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} z_i^2} \right) \\ - \exp\left( \frac{1}{D} \sum_{i=1}^{D} \cos(2\pi z_i) \right) + 20 + e \end{matrix}, \quad \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o},$$

$\boldsymbol{x} = [x_1, x_2, \cdots, x_D], o = [o_1, o_2, \cdots, o_D]$
where $\boldsymbol{x} \in [-32, 32]^D$, the global optimum $\boldsymbol{x}^* = \boldsymbol{o}$, and $F_6(\boldsymbol{x}^*) = 0$.

7. $F_7$: Shifted Rastrigin's Function

$$F_7(\boldsymbol{x}) = \sum_{i=1}^{D} \left( z_i^2 - 10\cos(2\pi z_i) + 10 \right), \quad \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o},$$

$\boldsymbol{x} = [x_1, x_2, \cdots, x_D], o = [o_1, o_2, \cdots, o_D]$
where $\boldsymbol{x} \in [-5.12, 5.12]^D$, the global optimum $\boldsymbol{x}^* = \boldsymbol{o}$, and $F_7(\boldsymbol{x}^*) = 0$.

8. $F_8$: Shifted Rotated Griewank's Function without Bounds

$$F_8(\mathrm{x}) = \frac{1}{4000} \sum_{i=1}^{D} z_i^2 - \prod_{i=1}^{D} \cos\left( \frac{z_i}{\sqrt{i}} \right) + 1, \mathbf{z} = \mathbf{x} - \mathbf{o}$$

$\boldsymbol{x} = [x_1, x_2, \cdots, x_D], o = [o_1, o_2, \cdots, o_D]$
where $\boldsymbol{x} \in [-600, 600]^D$, the global optimum $\boldsymbol{x}^* = \boldsymbol{o}$, and $F_8(\boldsymbol{x}^*) = 0$.

## 4.2. Parameter Setting

The parameter settings of all tested algorithms are as follows:

1. QFOA-1 and QFOA-2: set the population size as $M_{\text{osp}} = 10D$ and the maximum number of function evaluation as

$fes = 5000D$, as well as the parameters are set as $b_1 = 1$ and $b_2 = 0.5$ for both two QFOA models.

2. FOA-1 and FOA-2: the population size and the maximum number of function evaluation are set as $M_{osp} = 10D$ and $fes = 5000D$.

3. IFFO: set $M_{osp} = 10D$ and $fes = 5000D$; in addition, set parameters $\lambda_{max} = 0.5$ and $\lambda_{min} = 0.01$ according to the original literature [1].

4. MFOA: set $M_{osp} = 10D$ and $fes = 5000D$ as well as parameter $M = 10$ according to the original literature [24].

5. $\theta$-MAFOA: set $M_{osp} = 10D$ and $fes = 5000D$ as well as the $Cr = 0.9$ according to the original literature [13].

6. PSO: set the population size as $M_{pop} = 10D$ and the maximum number of function evaluation is $fes = 5000D$; other parameters are set as $c_1 = c_2 = 0.5$, $w_{max} = 0.9$ and $w_{max} = 0.4$ [2].

7. DE: set $M_{pop} = 10D$ and $fes = 5000D$; other parameters are set as $F = 0.6$ and $cr = 0.9$ [9].

8. ABC: set $M_{pop} = 10D$, $fes = 5000D$ and $Limit = 50$ [8].

All the above parameters values are set according to the suggestions in related literatures, where these values have been proved to lead to the good performance, empirically.

## 4.3. Comparison and Discussion

The mean best function values of the 50 independent trials for each combination of benchmark function and tested algorithm are recorded in Table 1, while the corresponding minimum best function values of the 50 independent trials are recorded in Table 2.

From Table 1, it can be seen that, in the average sense, two QFOA models show the great advantage in finding the global minimum on all sixteen benchmarks and are significantly superior to the basic standard FOA models on these functions. Compared with other improved FOA variants, two QFOA models achieve the significantly better performance on $F_1$-$F_3$ and $F_5$-$F_6$ with $D = 10$ as well as $F_1$ and $F_6$ with $D = 20$. For other functions, such as $F_7$-$F_8$ with $D = 10$ as well as $F_3$ and $F_5$-$F_8$ with $D = 20$, there is one QFOA model, either QFOA-1 or QFOA-2, that outperforms IFFO, MFOA, and $\theta$-MAFOA. It can be claimed that at least one FOA model performs the best on fifteen functions except for only $F_2$ with $D = 20$. Hence, in the average sense, it can be concluded that the proposed QFOA models do not only greatly improve the original standard FOA models, but also significantly exceed other improved FOA variants. Compared with PSO, DE, and ABC, Table 1 shows that both two QFOA models are superior to the three population-based algorithms on ten functions, while at least one QFOA is superior to the three population-based algorithms on all sixteen functions.

Table 2 indicates that the proposed QFOA models have the best performance than other tested algorithms on nine functions, and meanwhile, at least one QFOA outperforms the three population-based algorithms on fifteen functions except for only $F_2$ with $D = 20$, on which $\theta$-MAFOA achieves the best performance. Therefore, the proposed QFOA models have absolute superiority in dealing with complex continue optimization problems.

Moreover, to further study the performance of proposed QFOA models comparatively, the performance of all tested algorithms in dealing with benchmark functions are ranked according to the numerous results in Tables 1–2. Table 3 records the rank of mean results of tested algorithms. It can be seen that QFOA-1 wins total eleven first places and QFOA-2 wins six first places and six second places. As to the mean-rank, two QFOA models are in the top two ahead of other tested algorithms. For the rank of minimum results listed in Table 4, QFOA-1 wins total twelve first places and QFOA-2 wins seven first places and seven second places. Two QFOA models are still in the top two much more ahead of other tested algorithms according to the mean-rank. In a word, the proposed QFOA models surpassed the standard FOA models, some improved FOA variants and population-based algorithms.

In addition, the convergence curves of the ten tested algorithms on the eight benchmark functions are shown in Figure 1(a)-(h) for the dimension $D = 10$ and in Figure 2(a)-(h) for the dimension $D = 20$, respectively, where convergence curves show the mean best function values of 50 trials per iteration. Through comparing these convergence curves, it is easy to find that the QFOA models have the better convergence performance than other tested algorithms on most benchmark functions. QFOA-1 can converge much more rapidly during the early stage of the search process and then achieve the optimum so quickly. This characteristic is very remarkable on $F_1$-$F_2$, $F_4$-$F_6$ and $F_8$ with $D = 10$ as well as $F_1$, $F_4$-$F_6$ and $F_8$ with $D = 20$. On the contrary, QFOA-2 often shows its powerful search capability and accelerates convergence to the optimum during the later stage of the search process. Especially, QFOA-2 converges to the optimum on $F_3$, $F_4$, $F_7$ with $D = 10$ and 20, and converges to the sub-optimum on $F_1$-$F_2$ and $F_5$-$F_6$ with $D = 10$ as well as $F_1$ and $F_6$ with $D = 20$. The proposed QFOA models can not only find higher quality solutions but also possess faster convergence speed on benchmark functions.

Analyzing the experiment results, it is clear that, the proposed QFOA models perform significantly better than the standard FOA models, and meanwhile, are superior to or at least competitive with existing improved FOA variants as well as some population-based algorithms. Above all, the proposed QFOA models have faster convergence speed and stronger search capability, which could avoid most of the local optimum and significantly improve the shortcomings of the premature convergence and slow convergent speed during the later evolution process on the continuous function optimization problems.

# 5. QFOA MODELS FOR UAV PATH PLANNING

## 5.1. Model of Objective Function of UAV Path

The core work of path planning is to set up an effective objective or cost function which acts as the criterion to evaluate whether a path or a location of fruit fly is good or not [16]. The path planning problem is modeled as an optimization problem that minimizes the objective function. Assume the full UAV path is represented by a collection of discrete waypoints $\{p_0, p_1, ..., p_N, p_{N+1}\}$, where the coordinate of the waypoint $p_k$ is $(xp_k, yp_k, zp_k)$, $k = 0, ..., N+1$. The

Table 1 | Mean optimization results on benchmark functions.

| *D* | *F* | QFOA-1 | QFOA-2 | FOA-1 | FOA-2 | IFFO | MFOA | $\theta$-MAFOA | PSO | DE | ABC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | $F_1$ | $4.51\times10^{-29}$ | $2.47\times10^{-28}$ | $2.40\times10^{-1}$ | $3.78\times10^{-1}$ | $5.90\times10^{-5}$ | $3.70\times10^{-15}$ | $2.95\times10^{-10}$ | $8.14\times10^{-6}$ | $3.14\times10^{-6}$ | $2.25\times10^{-7}$ |
| | $F_2$ | $5.26\times10^{-14}$ | $6.78\times10^{-12}$ | $3.32\times10^{-1}$ | $3.70\times10^{-1}$ | $6.96\times10^{-1}$ | $3.25\times10^{-9}$ | $4.24\times10^{-9}$ | $2.42\times10^{-1}$ | 1.01 | 7.46 |
| | $F_3$ | $1.09\times10^{-12}$ | $9.97\times10^{-25}$ | $9.59\times10^{2}$ | $1.19\times10^{5}$ | 6.08 | $2.62\times10^{5}$ | $5.04\times10^{-6}$ | $8.03\times10^{-5}$ | $2.70\times10^{-4}$ | $4.58\times10^{-3}$ |
| | $F_4$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | $F_5$ | 4.26 | 4.47 | $2.89\times10^{1}$ | $1.35\times10^{3}$ | $2.82\times10^{1}$ | $2.08\times10^{1}$ | 9.00 | 9.36 | 6.78 | $3.26\times10^{1}$ |
| | $F_6$ | $2.32\times10^{-14}$ | $3.06\times10^{-14}$ | 1.45 | $1.57\times10^{1}$ | $2.76\times10^{-3}$ | $2.48\times10^{-8}$ | $2.20\times10^{-5}$ | $2.39\times10^{-4}$ | $1.14\times10^{-3}$ | $4.01\times10^{-1}$ |
| | $F_7$ | 3.104 | $1.58\times10^{-16}$ | $2.65\times10^{1}$ | $3.74\times10^{1}$ | $2.63\times10^{-5}$ | $1.45\times10^{1}$ | 4.85 | $1.07\times10^{1}$ | $2.13\times10^{1}$ | $1.60\times10^{-5}$ |
| | $F_8$ | $4.22\times10^{-17}$ | $6.22\times10^{-2}$ | $2.63\times10^{-2}$ | $4.26\times10^{1}$ | $3.71\times10^{-2}$ | $1.19\times10^{-1}$ | $3.01\times10^{-11}$ | $4.93\times10^{-1}$ | $4.96\times10^{-1}$ | $4.68\times10^{-2}$ |
| 20 | $F_1$ | $4.43\times10^{-20}$ | $2.94\times10^{-14}$ | 1.24 | 1.82 | $1.20\times10^{-4}$ | $9.25\times10^{-13}$ | $3.35\times10^{-9}$ | $8.34\times10^{-1}$ | $4.57\times10^{-5}$ | $8.86\times10^{-6}$ |
| | $F_2$ | $5.01\times10^{-4}$ | $1.02\times10^{-2}$ | 2.22 | 2.70 | $2.46\times10^{1}$ | $4.68\times10^{-4}$ | $2.07\times10^{-7}$ | 3.84 | $5.72\times10^{3}$ | $2.60\times10^{2}$ |
| | $F_3$ | $1.04\times10^{1}$ | $2.46\times10^{-8}$ | $1.03\times10^{4}$ | $7.37\times10^{5}$ | $1.42\times10^{1}$ | $1.31\times10^{6}$ | $3.94\times10^{-4}$ | $1.23\times10^{2}$ | $7.90\times10^{-3}$ | $1.41\times10^{-1}$ |
| | $F_4$ | 0 | 0 | $2.20\times10^{-1}$ | 1.52 | 0 | 0 | 0 | $8.00\times10^{-1}$ | 0 | 0 |
| | $F_5$ | $1.53\times10^{1}$ | $1.99\times10^{1}$ | $1.65\times10^{2}$ | $1.90\times10^{3}$ | $6.21\times10^{1}$ | $1.74\times10^{1}$ | $1.90\times10^{1}$ | $2.27\times10^{2}$ | $6.63\times10^{1}$ | $7.47\times10^{1}$ |
| | $F_6$ | $9.51\times10^{-11}$ | $1.55\times10^{-8}$ | 2.29 | $1.87\times10^{1}$ | $2.89\times10^{-3}$ | $3.71\times10^{-7}$ | $4.96\times10^{-5}$ | $3.87\times10^{-1}$ | $2.45\times10^{-3}$ | $1.04\times10^{1}$ |
| | $F_7$ | 8.68 | $1.06\times10^{-12}$ | $8.41\times10^{1}$ | $1.31\times10^{2}$ | $5.80\times10^{-5}$ | $5.42\times10^{1}$ | 9.43 | $2.04\times10^{1}$ | $8.28\times10^{1}$ | $6.51\times10^{-5}$ |
| | $F_8$ | $8.91\times10^{-15}$ | $4.37\times10^{-2}$ | $8.10\times10^{-2}$ | $5.82\times10^{-1}$ | $1.56\times10^{-2}$ | $8.82\times10^{-3}$ | $2.21\times10^{-10}$ | $8.92\times10^{-1}$ | $3.18\times10^{-1}$ | $3.27\times10^{-2}$ |

Note: MFOA, multi-swarm fruit fly optimization algorithm; QFOA, quantum-behaved fruit fly optimization algorithm; FOA, fruit fly optimization algorithm; IFFO, improved fruit fly optimization; $\theta$-MAFOA, angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism; PSO, particle swarm optimization; DE, differential evolution; ABC, artificial bee colony.

Table 2 | Minimum optimization results on benchmark functions.

| *D* | *F* | QFOA-1 | QFOA-2 | FOA-1 | FOA-2 | IFFO | MFOA | $\theta$-MAFOA | PSO | DE | ABC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | $F_1$ | 0 | $1.23\times10^{-32}$ | $8.03\times10^{-2}$ | $1.77\times10^{-1}$ | $6.80\times10^{-6}$ | $2.16\times10^{-16}$ | $1.67\times10^{-11}$ | $6.65\times10^{-14}$ | $5.58\times10^{-7}$ | $6.36\times10^{-9}$ |
| | $F_2$ | $1.97\times10^{-24}$ | $1.30\times10^{-15}$ | $1.53\times10^{-1}$ | $1.40\times10^{-1}$ | $1.99\times10^{-1}$ | $1.02\times10^{-10}$ | $6.97\times10^{-10}$ | $4.37\times10^{-3}$ | $2.31\times10^{-1}$ | $4.26\times10^{-1}$ |
| | $F_3$ | 0 | 0 | $3.41\times10^{2}$ | $2.15\times10^{4}$ | $1.41\times10^{-1}$ | $2.85\times10^{4}$ | $2.32\times10^{-7}$ | $5.64\times10^{-10}$ | $7.05\times10^{-5}$ | $1.12\times10^{-5}$ |
| | $F_4$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | $F_5$ | 2.38 | $2.50\times10^{-3}$ | $2.03\times10^{1}$ | $1.62\times10^{1}$ | $5.77\times10^{-2}$ | 3.46 | 9.00 | 5.25 | 3.71 | $1.37\times10^{-2}$ |
| | $F_6$ | $4.44\times10^{-15}$ | $4.44\times10^{-15}$ | $9.43\times10^{-1}$ | 1.32 | $1.28\times10^{-3}$ | $8.26\times10^{-8}$ | $8.99\times10^{-6}$ | $1.28\times10^{-6}$ | $4.39\times10^{-4}$ | $4.13\times10^{-4}$ |
| | $F_7$ | 0 | 0 | $1.66\times10^{1}$ | $1.99\times10^{1}$ | $5.37\times10^{-6}$ | 4.97 | $9.95\times10^{-1}$ | 3.98 | $1.18\times10^{1}$ | $9.26\times10^{-7}$ |
| | $F_8$ | 0 | $7.40\times10^{-3}$ | $1.06\times10^{-2}$ | $2.39\times10^{1}$ | $9.42\times10^{-5}$ | $3.45\times10^{-2}$ | $2.54\times10^{-12}$ | $3.34\times10^{-1}$ | $3.22\times10^{-1}$ | $7.28\times10^{-6}$ |
| 20 | $F_1$ | $1.05\times10^{-27}$ | $7.39\times10^{-20}$ | $7.58\times10^{-1}$ | 1.22 | $3.05\times10^{-5}$ | $7.51\times10^{-14}$ | $1.08\times10^{-9}$ | $3.83\times10^{-3}$ | $1.75\times10^{-5}$ | $2.28\times10^{-6}$ |
| | $F_2$ | $2.98\times10^{-7}$ | $3.11\times10^{-4}$ | 1.40 | 1.83 | 7.50 | $6.92\times10^{-5}$ | $9.20\times10^{-8}$ | 2.24 | $3.54\times10^{3}$ | $5.86\times10^{1}$ |
| | $F_3$ | $1.28\times10^{-18}$ | $6.73\times10^{-16}$ | $4.93\times10^{3}$ | $2.73\times10^{5}$ | $7.81\times10^{-1}$ | $4.38\times10^{5}$ | $9.50\times10^{-5}$ | 2.49 | $2.76\times10^{-3}$ | $5.12\times10^{-3}$ |
| | $F_4$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | $F_5$ | $1.26\times10^{1}$ | $2.20\times10^{-1}$ | $1.08\times10^{2}$ | $1.39\times10^{2}$ | $7.39\times10^{-1}$ | $1.22\times10^{1}$ | $1.90\times10^{1}$ | $4.45\times10^{1}$ | $3.15\times10^{1}$ | $3.67\times10^{-1}$ |
| | $F_6$ | $7.90\times10^{-14}$ | $3.66\times10^{-11}$ | 1.79 | 2.66 | $1.70\times10^{-3}$ | $1.13\times10^{-7}$ | $3.42\times10^{-5}$ | $3.35\times10^{-2}$ | $1.52\times10^{-3}$ | $1.80\times10^{-3}$ |
| | $F_7$ | 2.98 | $7.11\times10^{-15}$ | $6.50\times10^{1}$ | $9.72\times10^{1}$ | $2.11\times10^{-5}$ | $2.79\times10^{1}$ | 4.97 | 7.82 | $6.30\times10^{1}$ | $1.69\times10^{-5}$ |
| | $F_8$ | 0 | $1.22\times10^{-15}$ | $5.73\times10^{-2}$ | $1.46\times10^{1}$ | $1.12\times10^{-4}$ | $7.63\times10^{-11}$ | $6.54\times10^{-11}$ | $1.59\times10^{-1}$ | $1.12\times10^{-1}$ | $9.78\times10^{-6}$ |

MFOA, multi-swarm fruit fly optimization algorithm; QFOA, quantum-behaved fruit fly optimization algorithm; FOA, fruit fly optimization algorithm; IFFO, improved fruit fly optimization; $\theta$-MAFOA, angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism; PSO, particle swarm optimization; DE, differential evolution; ABC, artificial bee colony.

objective function of the UAV path can be defined as [31]

$$\min J = \sum_{i=1}^{5} f_i \tag{27}$$

where $f_i$, $i = 1,\ldots,5$, is the costs of flight length, safety, altitude, turning angle, climbing and gliding slope, and terrain constraint.

The length cost $f_1$ is defined as the sum of all path segment lengths from the first to last points:

$$f_1 = \sum_{k=0}^{N} l_k \tag{28}$$

with

$$l_k = \sqrt{\left(xp_{i+1} - xp_i\right)^2 + \left(yp_{i+1} - yp_i\right)^2 + \left(zp_{i+1} - zp_i\right)^2}$$

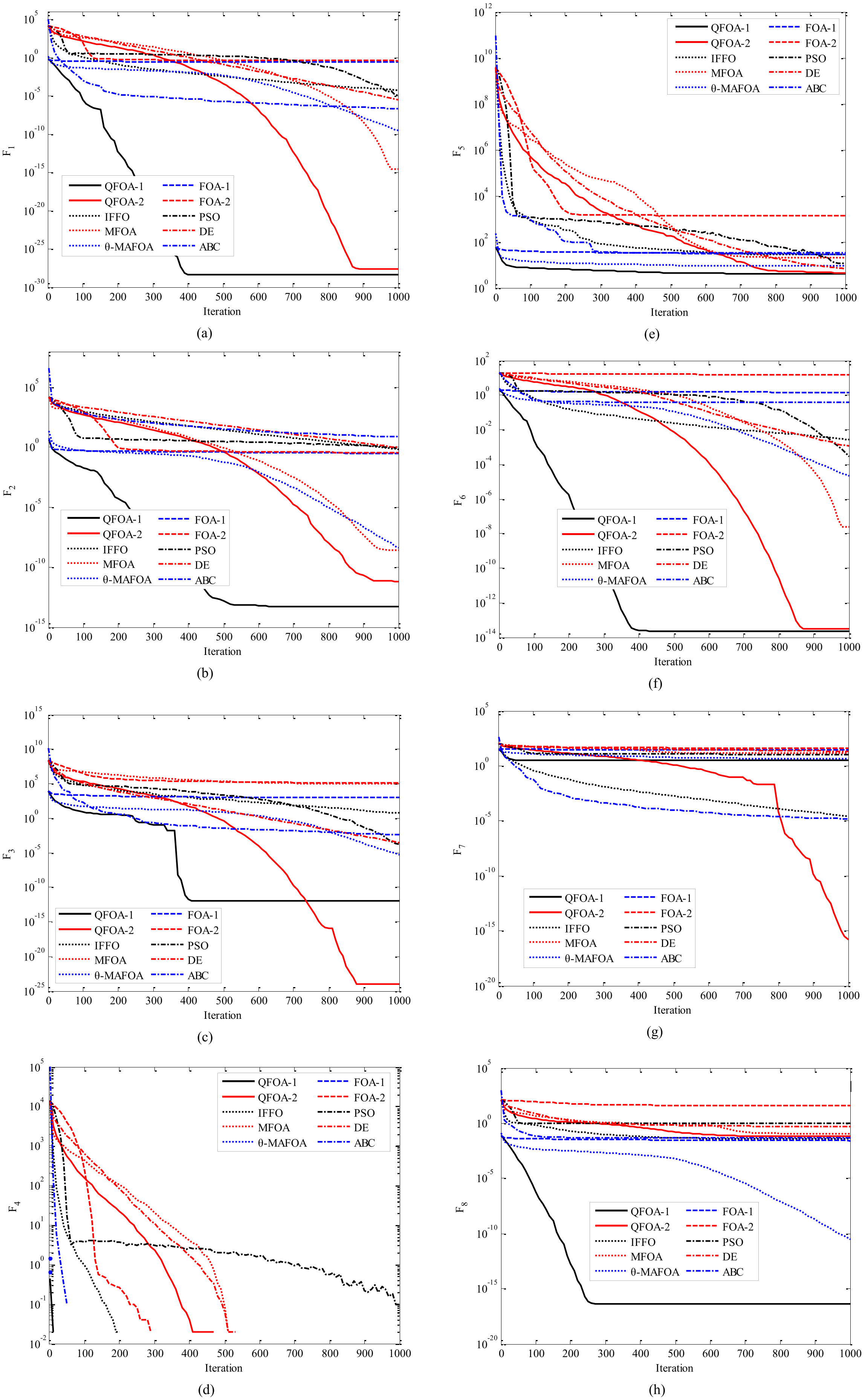
The threat cost $f_2$ penalizes the paths that fly dangerously close to threat sites, which depends on the length and the threat probability of each path segment as follows:

$$f_2 = \sum_{k=0}^{N} \left( l_k \cdot \sum_{j=1}^{n_T} P_{j,k} \right) \tag{29}$$

with

$$P_{j,k} = \begin{cases} 0 & d_{j,k} > R_{\max,j} \\ \dfrac{R_{\max,j}^4}{d_{j,k}^4 + R_{\max,j}^4} & d_{j,k} \le R_{\max,j} \end{cases}$$

where $P_{j,k}$ is the threat probability of the path segment $p_k p_{k+1}$ suffering from the $j$-th threat, $R_{\max,j}$ denotes the maximum threaten



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

**Figure 1.** Comparison of convergence curves on benchmark functions with $D = 10$.

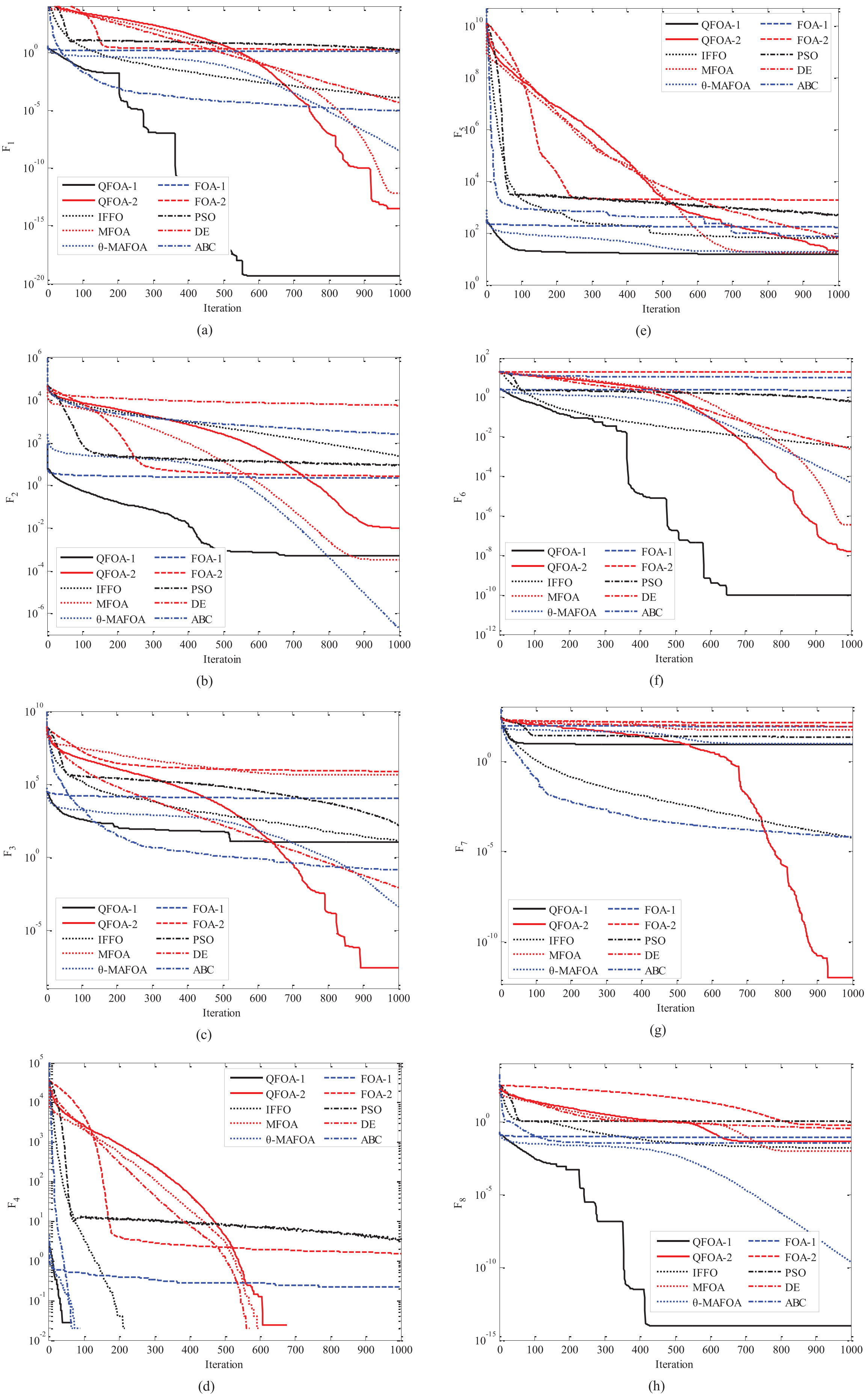


**Figure 2.** Comparison of convergence curves on benchmark functions with $D = 20$.

**Table 3** Rank of the mean optimization results.

| D | F | QFOA-1 | QFOA-2 | FOA-1 | FOA-2 | IFFO | MFOA | θ-MAFOA | PSO | DE | ABC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | $F_1$ | 1 | 2 | 9 | 10 | 8 | 3 | 4 | 7 | 6 | 5 |
| | $F_2$ | 1 | 2 | 6 | 7 | 8 | 3 | 4 | 5 | 9 | 10 |
| | $F_3$ | 2 | 1 | 8 | 9 | 7 | 10 | 3 | 4 | 5 | 6 |
| | $F_4$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | $F_5$ | 1 | 2 | 8 | 10 | 7 | 6 | 4 | 5 | 3 | 9 |
| | $F_6$ | 1 | 2 | 9 | 10 | 7 | 3 | 4 | 5 | 6 | 8 |
| | $F_7$ | 4 | 1 | 9 | 10 | 3 | 7 | 5 | 6 | 8 | 2 |
| | $F_8$ | 1 | 6 | 3 | 10 | 4 | 7 | 2 | 8 | 9 | 5 |
| 20 | $F_1$ | 1 | 2 | 9 | 10 | 7 | 3 | 4 | 8 | 6 | 5 |
| | $F_2$ | 3 | 4 | 5 | 6 | 8 | 2 | 1 | 7 | 10 | 9 |
| | $F_3$ | 5 | 1 | 8 | 9 | 6 | 10 | 2 | 7 | 3 | 4 |
| | $F_4$ | 1 | 1 | 8 | 10 | 1 | 1 | 1 | 9 | 1 | 1 |
| | $F_5$ | 1 | 4 | 8 | 10 | 5 | 2 | 3 | 9 | 6 | 7 |
| | $F_6$ | 1 | 2 | 8 | 10 | 6 | 3 | 4 | 7 | 5 | 9 |
| | $F_7$ | 4 | 1 | 9 | 10 | 2 | 7 | 5 | 6 | 8 | 3 |
| | $F_8$ | 1 | 6 | 7 | 9 | 4 | 3 | 2 | 10 | 8 | 5 |
| Mean-rank | | **1.8** | **2.4** | **7.2** | **8.8** | **5.3** | **4.4** | **3.1** | **6.5** | **5.9** | **5.6** |

MFOA, multi-swarm fruit fly optimization algorithm; QFOA, quantum-behaved fruit fly optimization algorithm; FOA, fruit fly optimization algorithm; IFFO, improved fruit fly optimization; θ-MAFOA, angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism; PSO, particle swarm optimization; DE, differential evolution; ABC, artificial bee colony.

**Table 4** Rank of the minimum optimization results.

| D | F | QFOA-1 | QFOA-2 | FOA-1 | FOA-2 | IFFO | MFOA | θ-MAFOA | PSO | DE | ABC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | $F_1$ | 1 | 2 | 9 | 10 | 8 | 3 | 5 | 4 | 7 | 6 |
| | $F_2$ | 1 | 2 | 7 | 6 | 8 | 3 | 4 | 5 | 9 | 10 |
| | $F_3$ | 1 | 1 | 8 | 9 | 7 | 10 | 4 | 3 | 6 | 5 |
| | $F_4$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | $F_5$ | 4 | 1 | 10 | 9 | 3 | 5 | 8 | 7 | 6 | 2 |
| | $F_6$ | 1 | 2 | 9 | 10 | 8 | 3 | 5 | 4 | 7 | 6 |
| | $F_7$ | 1 | 1 | 9 | 10 | 4 | 7 | 5 | 6 | 8 | 3 |
| | $F_8$ | 1 | 5 | 6 | 10 | 4 | 7 | 2 | 9 | 8 | 3 |
| 20 | $F_1$ | 1 | 2 | 9 | 10 | 7 | 3 | 4 | 8 | 6 | 5 |
| | $F_2$ | 2 | 4 | 5 | 6 | 8 | 3 | 1 | 7 | 10 | 9 |
| | $F_3$ | 1 | 2 | 8 | 9 | 6 | 10 | 3 | 7 | 4 | 5 |
| | $F_4$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | $F_5$ | 5 | 1 | 9 | 10 | 3 | 4 | 6 | 8 | 7 | 2 |
| | $F_6$ | 1 | 2 | 9 | 10 | 6 | 3 | 4 | 8 | 5 | 7 |
| | $F_7$ | 4 | 1 | 9 | 10 | 3 | 7 | 5 | 6 | 8 | 2 |
| | $F_8$ | 1 | 2 | 7 | 10 | 6 | 4 | 3 | 9 | 8 | 5 |
| Mean-rank | | **1.7** | **1.9** | **7.3** | **8.2** | **5.2** | **4.6** | **3.8** | **5.8** | **6.3** | **4.5** |

MFOA, multi-swarm fruit fly optimization algorithm; QFOA, quantum-behaved fruit fly optimization algorithm; FOA, fruit fly optimization algorithm; IFFO, improved fruit fly optimization; θ-MAFOA, angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism; PSO, particle swarm optimization; DE, differential evolution; ABC, artificial bee colony.

**Table 5** Cost values of the best UAV paths obtained by various algorithms on the 1st scenario.

| Algorithm | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ |
|---|---|---|---|---|---|
| QFOA-1 | **114.98** | 0 | 79.24 | 0 | 0 |
| QFOA-2 | 117.15 | 0 | **66.25** | 0 | 0 |
| FOA-1 | 118.89 | 0 | 132.28 | 0 | 0 |
| FOA-2 | 158.45 | 0 | 152.12 | 0 | 0 |

UAV, unmanned aerial vehicle; QFOA, quantum-behaved fruit fly optimization algorithm; FOA, fruit fly optimization algorithm.

range of the $j$-th threat, and $d_{j,k}$ is the distance between $p_k p_{k+1}$ and the $j$-th threat center.

The altitude cost $f_3$ is introduced to make the UAV fly at the low altitude, so that the UAV can benefit from the terrain mask effect. Meanwhile, $f_3$ can penalize the nonfeasible paths that pass through the terrain, as follows:

$$f_3 = \sum_{k=1}^{N} f_k^3 \tag{30}$$

with

$$f_k^3 = \begin{cases} C, & \text{if} \quad zp_k < H_{\text{map}}\left(xp_k, yp_k\right) + H_{\text{safe}} \\ zp_k, & \text{otherwise} \end{cases}$$

where $C$ is the penalty factor, $H_{\text{map}}(xp_k, yp_k)$ is the ground level at the point $(xp_k, yp_k)$, and $H_{\text{safe}}$ is the minimal safe flight height.

The lateral maneuvering cost $f_4$ is the sum of those points that do not satisfy the turning constraint as follows:

$$f_4 = \sum_{k=1}^{N} f_k^4 \tag{31}$$

with

$$f_k^4 = \begin{cases} 0, & \text{if} \quad \phi_k \le \phi_k^{\max} \\ C, & \text{otherwise} \end{cases}$$

where $\phi_k$ denotes the turning angle at the waypoint $p_k$, and $C$ is the penalty factor. The UAV maneuverability is constrained by the maximum turning angle $\phi_k{}^{\max}$ as follows:

$$\phi_k^{\max} = \frac{n_{\max} g}{V^2} \cdot \sqrt{\left(xp_{k+1} - xp_k\right)^2 + \left(yp_{k+1} - yp_k\right)^2} \tag{32}$$

where $g$ is the gravitational acceleration, $n_{\max}$ is the maximum lateral overload and $V$ is the flight velocity.

The vertical maneuvering cost $f_5$ is the sum of those points that do not satisfy the slope constraint as follows:

$$f_5 = \sum_{k=1}^{N} f_k^5 \tag{33}$$

with

$$f_k^5 = \begin{cases} 0, & \text{if} \quad \beta_k \le s_k \le \alpha_k \\ C, & \text{otherwise} \end{cases}$$

where $s_k$ is the UAV slope at the waypoint $p_k$, $\alpha_k$ is the maximum climbing slope and $\beta_k$ is the minimum gliding slope, which are calculated as follows [4,16]:

$$\begin{aligned} \alpha_k &= -1.5377 \times 10^{-10} zp_k^2 - 2.6997 \times 10^{-5} zp_k + 0.4211 \\ \beta_k &= 2.5063 \times 10^{-9} zp_k^2 - 6.3014 \times 10^{-6} zp_k - 0.3257 \\ s_k &= \frac{zp_{k+1} - zp_k}{\sqrt{\left(xp_{k+1} - xp_k\right)^2 + \left(yp_{k+1} - yp_k\right)^2}} \end{aligned} \tag{34}$$

## 5.2. Path Representation in 3D Environment

The full UAV path is represented by a collection of $N$ discrete waypoints except the given start and target points. Considering each waypoint $p_k$, $k = 1, \dots, N$, are determined by three coordinates ($xp_k$, $yp_k$, $zp_k$), the total dimension of the decision variable in a UAV path is $3N$. In order to reduce the problem dimension and meanwhile to obtain the smooth path, the B-Spline-based path representation strategy is adopted in this paper to determine these waypoints and then represent the desired UAV path.

Given $n$+1 control points with the coordinates ($x_0$, $y_0$, $z_0$), ..., ($x_n$, $y_n$, $z_n$), the B-Spline curve can be constructed by [31]

$$\begin{aligned} X(t) &= \sum_{i=0}^{n} x_i \cdot B_{i,K}(t) \\ Y(t) &= \sum_{i=0}^{n} y_i \cdot B_{i,K}(t) \\ Z(t) &= \sum_{i=0}^{n} z_i \cdot B_{i,K}(t) \end{aligned} \tag{35}$$

where $B_{i,K}(t)$ is defined recursively as follows:

$$B_{i,1}(t) = \begin{cases} 1, & \mathit{if} \quad Knot(i) \le t < Knot(i+1) \\ 1, & \mathit{if} \quad Knot(i) \le t \le Knot(i+1)\, and t = n - K + 2 \\ 0, & \mathit{otherwise} \end{cases} \tag{36}$$

$$B_{i,K}(t) = \frac{(t - Knot(t)) \cdot B_{i,K}(t)}{Knot(i+K-1) - Knot(i)} + \frac{(Knot(i+K) - t) \cdot B_{i+1,K-1}(t)}{Knot(i+K) - Knot(i+1)} \tag{37}$$

with

$$\begin{aligned} &Knot(i) = 0, && \text{if} \quad i < K \\ &Knot(i) = i - K + 1, && \text{if} \quad K \le i \le n \\ &Knot(i) = n - K + 2, && \text{if} \quad n < i \end{aligned}$$

where the order $K$ is used to control the curve smoothness. The parameter $t$ varies from 0 to $n$–$K$+2 with a constant step, providing the discrete points of the B-Spline curve.

Using the B-Spline-based path representation, if the UAV path is represented by the collection of discrete waypoints {$p_0$, $p_1$, ..., $p_N$, $p_{N+1}$} with the coordinates ($xp_k$, $yp_k$, $zp_k$), $k = 1, \dots, N$, for each waypoint, then the corresponding control point collection is {$w_0$, $w_1$, ..., $w_n$, $w_{n+1}$} with the coordinates ($xc_i$, $yc_i$, $zc_i$), $i = 1, \dots, n$. Both the first waypoint $p_0$ and the first control point $w_0$ are the given start point, while both the last waypoint $p_{N+1}$ and the last control point $w_{n+1}$ are the given target point. The task of the path planner is to generate $n$ control points, and thus, these control points can achieve the B-Spline curve-based path together with the given start and target points.

## 5.3. Path Planning Using QFOA

In order to obtain the optimal UAV flight path, the B-Spline curve-based path connecting the given start and target points should have the minimum value of cost function. Using the proposed QFOA models to design the UAV path planner, the coordinates of the control points are treated as the food source of fruit fly swarm. The number of control points is $n$, and then, $D = 3n$ parameters {$x_1$, $x_2$, ..., $x_D$} should be determined by the path planner, where ($x_i$, $x_{n+i}$, $x_{2n+i}$), $i = 1,2,\dots,n$, denotes the 3D coordinates of the $i$-th control point $w_i$. That is to say, ($xc_i$, $yc_i$, $zc_i$) = ($x_i$, $x_{n+i}$, $x_{2n+i}$). These coordinates take values within the limits of the constrained physics 3D space of the mission map. Each candidate path obtained by path planner is evaluated according to the cost function described in (28), which is employed as the smell concentration judgment function for fruit flies. Suppose that the mission space of the UAV flight path is limited in $[0,L_{\max}]\times[0,W_{\max}]$, and the maximum flight altitude is $H_{\max}$. The following section will show how to apply QFOA models to the path planning problem:

1. Procedure of path planning using QFOA-1

At the $t$-th iteration of QFOA-1, locations of the $i$-th fruit fly is represented by ($X_{i,1}$,$X_{i,2}$,...,$X_{i,D}$) and ($Y_{i,1}$,$Y_{i,2}$,...,$Y_{i,D}$), and the judged value of smell concentration ($S_{i,1}$,$S_{i,2}$,...,$S_{i,D}$) corresponds to the control point coordinates ($xc_1$,...,$xc_n$, $yc_1$,...,$yc_n$, $zc_1$,...,$zc_n$). The detailed implemented procedure of QFOA-1 applied to UAV path planning is described in Algorithm 5.

2. Procedure of path planning using QFOA-2

For the $i$-th fruit fly at the $t$-th iteration in the QFOA-2, both the location and the judged value of smell concentration are represented by $(x_{i,1},x_{i,2},...,x_{i,D})$, which are treated as the decision variable of the optimization problem. Therefore, for the QFOA-2-based path planning problem, the location of the fruit fly $(x_{i,1},x_{i,2},...,x_{i,D})$ corresponds to the control point coordinates $(xc_1,...,xc_n,\ yc_1,...,yc_n,\ zc_1,...,zc_n)$, and thus can be decoded into a candidate UAV path using the B-Spline curve. The implemented procedure of QFOA-2 applied to UAV path planning is described in Algorithm 6.

**Algorithm 5:** Procedure of path planning using QFOA-1

```
/* Mission */
Mission information: start and target points $(x_S,y_S)^T$, $(x_T,y_T)^T$,
                     UAV performance parameters,
                     the number of the control points $n$..
Threat information: threat centres $(x_{threat\ j}, y_{threat\ j})^T$,
                    threat radius $R_j$.
Map information: the mission map range $L_{max,}$ $W_{max}$, $H_{max}$.
/* Initialization */
Set the generation counter $g = 0$.
Generation counter $g = g + 1$;
Randomly initialize the fruit fly swarm's location as $[X_{axis}, Y_{axis}]$ and
  ensure the judged value of smell concentration $S$ is within the search
  range $[0, 1]^D$.
/* Iterative search */
while termination criteria is not satisfied do
  Generation counter $g = g + 1$.
  Calculate the parameter $b = b_1$ logsig(10 (0.5 −g/$G_{max}$)) + $b_2$.
  /* Quantum-behaved search using osphresis */
  for $i = 1: M_{osp}$ do
    for $j = 1: D$ do
      Calculate the characteristic length:
        $L_{x,i,j} = 2b|X_{axis} - X_{i,j}|$, $L_{y,i,j} = 2b|Y_{axis} - Y_{i,j}|$.
      while 1 do
        Generate the random direction and distance as follows:
        if random > 0.5 then
          $X_{i,j} = X_{axis,j} + 0.5\ L_{x,i,j} \ln(1/r_x)$;
        else
          $X_{i,j} = X_{axis,j} - 0.5\ L_{x,i,j} \ln(1/r_x)$.
        if random > 0.5 then
          $Y_{i,j} = Y_{axis,j} + 0.5\ L_{y,i,j} \ln(1/r_y)$
        else
          $Y_{i,j} = Y_{axis,j} - 0.5\ L_{y,i,j} \ln(1/r_y)$.
        Calculate the judged value of smell concentration:
        $S_{i,j} = 1/\sqrt{X_{i,j}^2 + Y_{i,j}^2}$.
        if $S_{i,j} \le 1$ then
          break.
      end while
  end for
  /* Evaluation */
  Set $(xc_0, yc_0, zc_0)^T$ as the coordinates of the start point.
  for $j = 1: n$ do
      $xc_j = S_{i,j} \times L_{max}$
      $yc_j = S_{i,n+j} \times W_{max}$
      $zc_j = S_{i,2n+j} \times H_{max}$
  end for
  Set $(xc_{n+1}, yc_{n+1}, zc_{n+1})^T$ as the coordinates of the target point.
  Generate the B-Spline curve using the control points $(xc_0, yc_0, zc_0)^T$, ...,
    $(xc_{n+1}, yc_{n+1}, zc_{n+1})^T$.
  Evaluate smell concentrations (the cost value) according to (27):
    $Smell_i = J(Path_i)$, $Path_i = [(xc_0,yc_0,zc_0)^T,...,(xc_{n+1},yc_{n+1},zc_{n+1})^T]$.
  end for
  /* Search using vision */
  Select the fruit fly that has the best smell concentration:
    $index = \arg\max(Smell_i)$.
  if $[X_{index}, Y_{index}]$ has the better smell than $[X_{axis}, Y_{axis}]$, then
      $X_{axis} = X_{index}$ and $Y_{axis} = Y_{index}$.
end while
/*Output*/
Decode and output the best path for UAV.
```

**Algorithm 6:** Procedure of path planning using QFOA-2

```
Mission information: start and target points $(x_S,y_S)^T$, $(x_T,y_T)^T$,
                     UAV performance parameters,
                     the number of the control points $n$.
Threat information: threat centres $(x_{threat j}, y_{threat j})^T$,
                    threat radius $R_j$.
Map information: the mission map range $L_{max,}$ $W_{max}$, $H_{max}$.
/* Initialization */
Set the generation counter $g = 0$.
Set the number of fruit flies as $M_{osp}$ as well as parameters $b_1$, $b_2$.
Randomly initialize $\boldsymbol{x}_{axis} = [x_{axis,1}, x_{axis,2}, ..., x_{axis,D}]$ with $[0, 1]^D$.
/* Iterative search */
while termination criteria is not satisfied do
  Generation counter $g = g + 1$.
  Calculate the parameter $b = b_1$ logsig(10 (0.5 −g/$G_{max}$)) + $b_2$.
  /* Quantum-behaved search using osphresis */
  for $i = 1: M_{osp}$ do
    for $j = 1: D$ do
      Calculate the characteristic length:
        $L_{ij} = 2b|X_{axis,j} - X_{ij}|$.
      while 1 do
        Generate the random direction and distance as follows:
        if random > 0.5 then
          $X_{i,j} = X_{axis,j} + 0.5\ L_{i,j} \ln(1/r_x)$;
        else
          $X_{i,j} = X_{axis,j} - 0.5\ L_{i,j} \ln(1/r_x)$.
        if random > 0.5 then
          break
      end while
    end for
    /* Evaluation */
    Set $(xc_0, yc_0, zc_0)^T$ as the coordinates of the start point.
    for $j = 1: n$ do
        $xc_j = x_{i,j} \times L_{max}$
        $yc_j = x_{i,n+j} \times W_{max}$
        $zc_j = x_{i,2n+j} \times H_{max}$
    end for
    Set $(xc_{n+1}, yc_{n+1}, zc_{n+1})^T$ as the coordinates of the target point.
    Generate the B-Spline curve using the control point set
  $(xc_0, yc_0, zc_0)^T$, ..., $(xc_{n+1}, yc_{n+1}, zc_{n+1})^T$.
    Evaluate smell concentrations (the cost value) according to (27):
  $Smell_i = J(Path_i)$, $Path_i = [(xc_0,yc_0,zc_0)^T,..., (xc_{n+1},yc_{n+1},zc_{n+1})^T]$.
  end for
  /* Search using vision */
  Select the fruit fly that has the best smell concentration:
  $index = \arg\max(Smell_i)$.
  if $[X_{index}, Y_{index}]$ has the better smell than $[X_{axis}, Y_{axis}]$, then
      $X_{axis} = X_{index}$ and $Y_{axis} = Y_{index}$.
end while
  /*Output*/
Decode and output the best path for UAV.
```

## 6. EXPERIMENT RESULTS

To evaluate our designed QFOA-based path planners, several comparison simulations are carried out. We mainly concern the performance of the proposed QFOA algorithms versus the original standard FOA models, and thus, the QFOA-1, QFOA-2, FOA-1, and FOA-2 are tested under two scenarios in this section. All the tested algorithms are coded using Matlab-2009a on a personal computer. Due to the randomness nature of swarm intelligence algorithms, every tested algorithm is run 50 times independently for each scenario and the statistical results are used for evaluation and comparison. The mission map is a rectangular region with known terrain and threat environments. The number of the B-Spline control points is set as $n = 4$, which means the dimension of the problem is $D = 3n = 12$. Other parameters of the UAV are set as the maximum lateral overload $n_{\max} = 5$, the minimum safety flight altitude $H_{\text{safe}} = 50$ m, and the flight velocity $V = 200$ m/s. To make a fair comparison, the same maximum iteration number is used for all algorithms as the stopping criterion, while the population size is set as $M_{\text{osp}} = 40$ for all algorithms.

Figures 3 and 4 show the simulation results obtained by QFOA-1, QFOA-2, FOA-1, and FOA-2 on two different scenarios during 50 independent runs. The best UAV paths obtained by the four algorithms on two different scenarios are projected on the *xoy* contour plane as shown in Figures 3(a) and 4(a), respectively. In the maps, the circle dot “∘” in the lower-left corner is the start point, the square dot “□” in the upper-right corner is the target point, and the gray round areas represent the threat regions. As is shown, all algorithms succeed in finding the safe path that completely avoids all the threat regions. The change curves of the flight altitude of the best UAV paths along the length for the tested algorithms are displayed in Figures 3(b) and 4(b). Values of the five cost of the obtained best UAV paths for the 1st and 2nd scenarios are compared in Tables 5 and 6. It can be seen that all tested algorithms can achieve the feasible paths that satisfy all threat and altitude constraints, while QFOA-1 and QFOA-2 generates the shorter and lower paths than the original FOA algorithms. The proposed QFOA-based path planner can generate the UAV path with the threat-avoidance and terrain-following characteristics, and thus, increases both the survivability of the aircraft and the effectiveness of mission. Comparing QFOA-1 and QFOA-2, it can be seen that the cost criteria of the best UAV path obtained by QFOA-2 are smaller than those by QFOA-1, which indicates the powerful searching ability of QFOA-2.

The statistical results of four tested algorithms during 50 runs on two test scenarios are listed in Tables 7 and 8, where the best, median, mean, worst, and standard deviation of the cost value are recorded. The best values of each column are highlighted using boldface. The search ability and stability of a swarm intelligence

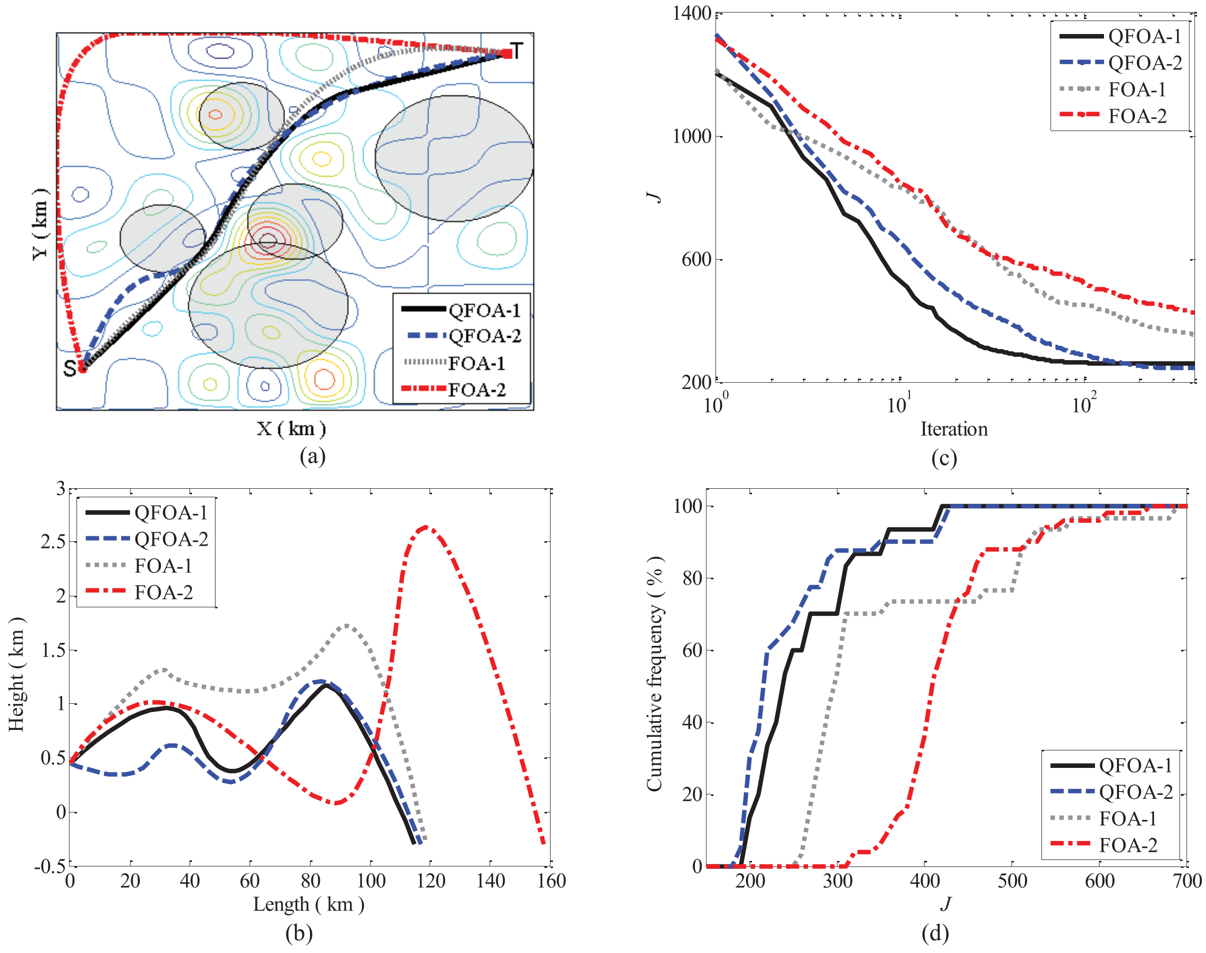


**Figure 3.** Results comparison on the 1st test scenario, including (a) best paths, (b) flight height of the best paths along the path length, (c) average convergence curves, and (d) cumulative frequency graphs of the cost values.

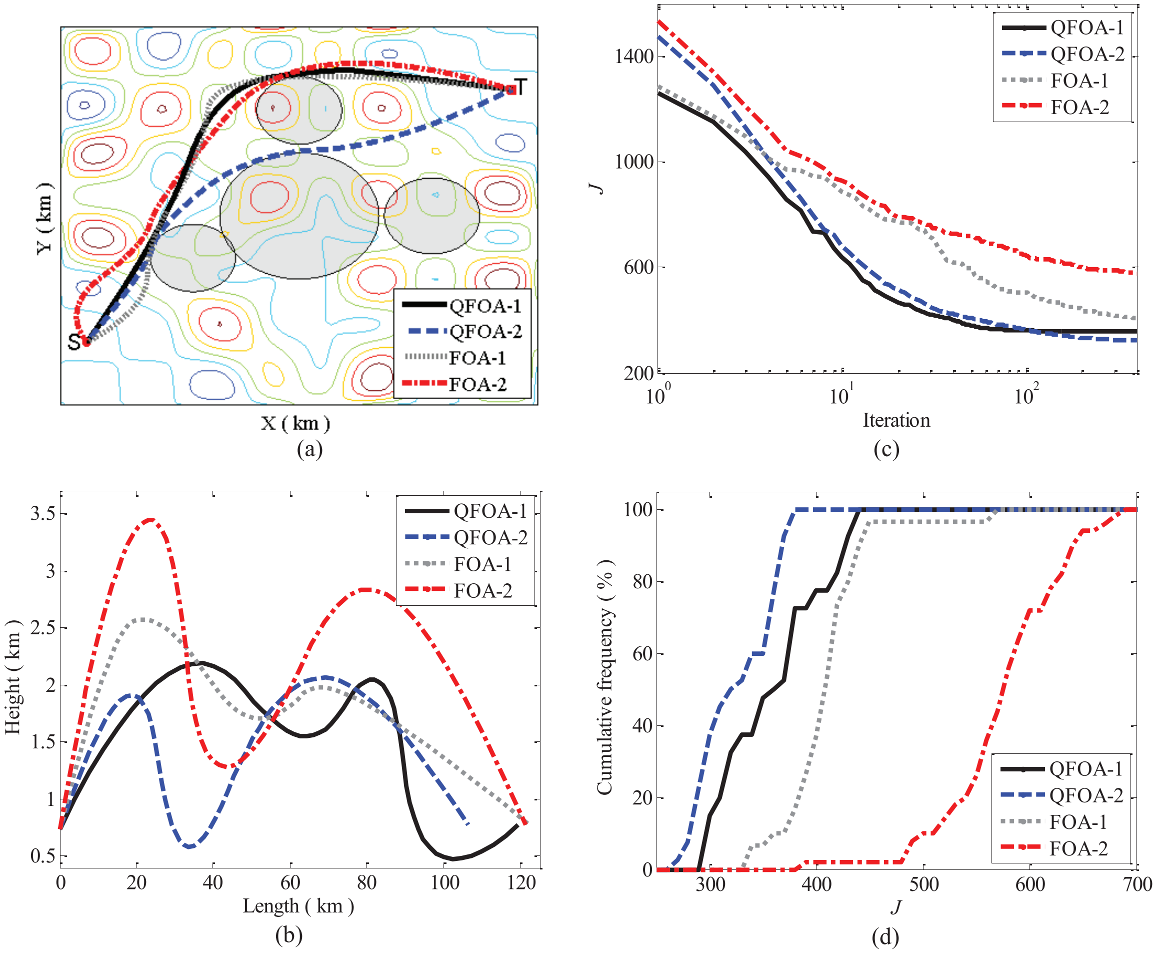


**Figure 4.** Results comparison on the 2nd test scenario, including (a) best paths, (b) flight height of the best paths along the path length, (c) average convergence curves, and (d) cumulative frequency graphs of the cost values.

algorithm can be reflected by the mean cost value and the standard deviation, respectively. Compared these statistical results, it is observed that QFOA-2 obtains the minimum best, mean, median, and worst cost, which indicates QFOA-2 has the most powerful optimization ability in the statistical sense. QFOA-1 also can obtain the smaller value of all statistical results than the original standard FOA models. The convergence curves of the average best cost values are displayed in Figures 3(c) and 4(c). It can be seen that, QFOA models achieve the faster convergence speed and the smaller cost than FOA models. The global convergence performance can be sorted as QFOA-2 > QFOA-1 > FOA-1 > FOA-2.

Furthermore, in order to study the distribution of the solutions, the cumulative frequency graphs of the minimum cost values during independent runs are listed in Figures 3(d) and 4(d), where the horizontal axis denotes the cost value and vertical axis denotes the cumulative frequency. Define the cumulative frequency as follows:

$$P_{\mathrm{cum}}(x) = \frac{N(J_{\min} \leq x)}{50} \times 100\% \tag{38}$$

where $N(J_{\min} < x)$ is the number of runs that obtain the cost value smaller than $x$. The result figures clearly show that both QFOA models perform better than original standard FOA. For example, on the 1st test scenario, all cost values obtained by QFOA are distributed in the interval smaller than 450, while only about 70% of FOA-1 trials and 80% of FOA-2 trials can achieve the same performance. As to the 2nd test scenario, 100% of cost values obtained by QFOA-2 are smaller than 350, while 100% of QFOA-1 trials, about 95% of FOA-1 trials and only 5% of FOA-2 trials achieve cost values less than 450. These experimental results demonstrate that QFOAs maintain the higher performance than the original standard FOA models in solving the UAV path planning problem, and meanwhile, QFOA-2 shows the better performance than QFOA-1. Above all, the proposed QFOA models have the advantage over the original standard FOAs in terms of searching ability, stability, and robustness.

**Table 6** Cost values of the best UAV paths obtained by various algorithms on the 2nd scenario.

| Algorithm | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ |
|---|---|---|---|---|---|
| QFOA-1 | 119.68 | 0 | 177.04 | 0 | 0 |
| QFOA-2 | **106.56** | 0 | **156.19** | 0 | 0 |
| FOA-1 | 122.41 | 0 | 214.62 | 0 | 0 |
| FOA-2 | 121.82 | 0 | 264.83 | 0 | 0 |

UAV, unmanned aerial vehicle; QFOA, quantum-behaved fruit fly optimization algorithm; FOA, fruit fly optimization algorithm.

**Table 7** | Performance comparison on the 1st test scenario.

| Algorithm | Best | Median | Mean | Worst | Std. |
|---|---|---|---|---|---|
| QFOA-1 | 194.22 | 236.36 | 259.84 | **418.38** | **63.70** |
| QFOA-2 | **183.40** | **215.79** | **244.64** | 429.97 | 69.14 |
| FOA-1 | 251.17 | 297.32 | 351.73 | 682.51 | 117.44 |
| FOA-2 | 310.57 | 407.73 | 423.55 | 652.89 | 65.31 |

QFOA, quantum-behaved fruit fly optimization algorithm; FOA, fruit fly optimization algorithm.

**Table 8** | Performance comparison on the 2nd test scenario.

| Algorithm | Best | Median | Mean | Worst | Std. |
|---|---|---|---|---|---|
| QFOA-1 | 296.72 | 360.20 | 328.21 | 437.77 | 46.92 |
| QFOA-2 | **262.75** | **323.90** | **325.51** | **319.19** | **35.57** |
| FOA-1 | 337.03 | 407.66 | 408.49 | 564.44 | 40.28 |
| FOA-2 | 386.65 | 572.48 | 574.79 | 689.98 | 55.86 |

QFOA, quantum-behaved fruit fly optimization algorithm; FOA, fruit fly optimization algorithm.

## 7. CONCLUSIONS

This paper introduces the quantum theory into FOA model and replaces the osphresis-based search of FOA as the quantum behavior-based searching mechanism, and thus, proposed the QFOA variant. In QFOA models, the Delta potential well is established around the fruit fly swarm location and fruit flies find and move toward the food source following the rules of wave function. Several comparative studies are conducted based on benchmark functions. Results show the proposed QFOA models are superior to the original standard FOA models and other improved FOA variants, as well as several state-of-the-art population-based algorithms, such as DE, PSO, and ABC. The proposed QFOA show the powerful searching abilities, robustness, and global convergence in solving continuous optimization problems.

In addition, the proposed QFOA models are also successfully applied to solve UAV path planning problem and the QFOA-based path planner are designed. The detailed implemented procedures of the path planning are given. Simulation results on two scenarios show that the proposed QFOA can solve this complex optimization problem and achieve much better performance in terms of searching ability, stability, and robustness than the original standard FOA models, which further indicates that the proposed QFOA is the powerful optimization technique.

It is the first work that take advantage of quantum theory to enhance the original FOA. Another contribution of this paper is that the improvement strategy is simultaneously employed to two standard FOA models. In the future work, we will further investigate the performance of the QFOA models and apply the them to the multi-objective optimization problems.

## ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China (No. 61703012), Beijing Natural Science Foundation (No. 4182010).

## REFERENCES

[1] Q.K. Pan, H.Y. Sang, J.H. Duan, L. Gao, An improved fruit fly optimization algorithm for continuous function optimization problems, Knowledge Based Syst. 62 (2014), 69–83.

[2] Y.W. Zhang, G.M. Cui, J.T. Wu, W.T. Pan, Q. He, A novel multi-scale cooperative mutation fruit fly optimization algorithm, Knowledge Based Syst. 114 (2016), 24–35.

[3] R. S. Patwal, N. Narang, H. Garg, A novel TVAC-PSO based mutation strategies algorithm for generation scheduling of pumped storage hydrothermal system incorporating solar units, Energy. 142 (2018), 822–837.

[4] X.Y. Zhang, H.B. Duan, An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning, Appl. Soft Comput. 26 (2015), 270–284.

[5] H. Garg, A hybrid PSO-GA algorithm for constrained optimization problems, Appl Mathematics Comput. 274 (2016), 292–305.

[6] W. Gao, New Ant colony optimization algorithm for the traveling salesman problem, Int. J. Comput. Intell. Syst. 13 (2020), 44–55.

[7] Y.G. Fu, M.Y. Ding, C.P. Zhou, Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV, IEEE Trans. Syst. Man. Cybern. A. Syst. Hum. 42 (2012), 511–526.

[8] B. Kruekaew, W. Kimpan, Enhancing of artificial bee colony algorithm for virtual machine scheduling and load balancing problem in cloud computing, Int. J. Comput. Intell. Syst. 13 (2020), 496–510.

[9] I. Brajevic, P. Stanimirovic, An improved chaotic firefly algorithm for global numerical optimization, Int. J. Comput. Intell. Syst. 12 (2018), 131–148.

[10] X.Y. Cui, Z. Chen, F.L. Yin, Differential evolution and local search based monarch butterfly optimization algorithm with applications, Int. J. Comput. Intell. Syst. 12 (2018), 149–163.

[11] H. Garg, A hybrid GSA-GA algorithm for constrained optimization problems, Inf. Sci. 478 (2019), 499–523.

[12] H. Li, W Gan, A decomposition-based multiobjective chemical reaction optimization algorithm for community detection in complex networks, Int. J. Comput. Intell. Syst. 13 (2020), 524–537.

[13] H. Garg, An efficient biogeography based optimization algorithm for solving reliability optimization problems, Swarm Evol. Comput. 24 (2015), 1–10.

[14] H. Garg, A Hybrid GA-GSA Algorithm for Optimizing the Performance of an Industrial System by Utilizing Uncertain Data, Handbook of Research on Artificial Intelligence Techniques and Algorithms, IRM Press, 2015.

[15] W.T. Pan, A new fruit fly optimization algorithm: taking the financial distress model as an example, Knowledge Based Syst. 26 (2012), 69–74.

[16] X.Y. Zhang, X.Y. Lu, S.M. Jia, X.Z. Li, A novel phase angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism applied to UAV path planning, Appl. Soft Comput. 70 (2018), 371–388.

[17] X.Y. Zhang, S.M. Jia, X.Z. Li, M. Jian, Design of the fruit fly optimization algorithm based path planner for UAV in 3D environments, in: Proceedings of 2017 IEEE International Conference on Mechatronics and Automation, August 6–9, Takamatsu, Japan, 2017, pp. 381–386.

[18] J.Q. Li, Q.K. Pan, K. Mao, A hybrid fruit fly optimization algorithm for the realistic hybrid flowshop rescheduling problem in steelmaking systems, IEEE Trans. Automation Sci. Eng. 13 (2016), 932–949.

[19] T. Meng, Q.K. Pan, An improved fruit fly optimization algorithm for solving the multidimensional knapsack problem, Appl. Soft Comput. 50 (2017), 79–93.

[20] X.F. Yuan, Y.M. Liu, Y.Z. Xiang, X.G. Yan, Parameter identification of BIPT system using chaotic-enhanced fruit fly optimization algorithm, Appl. Mathematics Comput. 268 (2015), 1267–1281.

[21] W. Sheng, Y. Bao, Fruit fly optimization algorithm based fractional order fuzzy-PID controller for electronic throttle, Nonlinear Dynamics. 73 (2013), 611–619.

[22] S. Kanarachos, J. Griffin, M.E. Fitzpatrick, Efficient truss optimization using the contrast-based fruit fly optimization algorithm, Comput. Struct. 182 (2017), 137–148.

[23] M. Mitić, N. Vuković, M. Petrović, Z. Miljković, Chaotic fruit fly optimization algorithm, Knowledge Based Syst. 89 (2015), 446–458.

[24] X.F. Yuan, X.S. Dai, J.Y. Zhao, Q. He, On a novel multi-swarm fruit fly optimization algorithm and its application, Appl. Mathematics Comput. 233 (2014), 260–271.

[25] L. Wu, W.S. Xiao, L. Zhang, Q. Liu, J.L. Wang, An improved fruit fly optimization algorithm based on selecting evolutionary direction intelligently, Int. J. Comput. Intell. Syst. 9 (2016), 80–90.

[26] X.Y. Zhang, S. Xia, Quantum behaved fruit fly optimization algorithm for continuous function optimization problems, in: Proceedings of the 9th International Conference on Swarm Intelligence, July 26–30, Chiengmai, Thailand, 2019, pp. 331–340.

[27] L. Gyongyosi, S. Imre, A survey on quantum computing technology, Comput. Sci. Rev. 31 (2019), 51–71.

[28] J. Sun, B. Feng, W. Xu, Particle swam optimization with particles having quantum behavior, in: Proceedings of IEEE conference on cybernetics and intelligent systems, 2004, pp. 111–116.

[29] T.Y. Liu, L.C. JiaoW.P. Ma, R.H. Shang, Quantum-behaved particle swarm optimization with collaborative attractors for nonlinear numerical problems, Commun. Nonlinear Sci. Numerical Simulation. 44 (2017), 167–183.

[30] F.B.Ozsoydan, A.Baykasoglu, Quantum firefly swarms for multimodal dynamic optimization problems, Expert Syst. Appl. 115 (2019), 189–199.

[31] E. Besada-Portas, L. de la Torre, A. Moreno, J. L. Risco-Martín, *On the performance comparison of multi-objective evolutionary UAV path planners*, Inf. Sci. **238** (2013), 111–125.