

Research Article

Three-Dimensional Path Planning Method for Autonomous Underwater Vehicle Based on Modified Firefly Algorithm

Chang Liu, Yuxin Zhao, Feng Gao, and Liqiang Liu

College of Automation, Harbin Engineering University, Harbin, Heilongjiang 150001, China

Correspondence should be addressed to Chang Liu; liuchang407@hrbeu.edu.cn

Received 22 October 2014; Accepted 23 December 2014

Academic Editor: Xin-She Yang

Copyright © 2015 Chang Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Path planning is a classic optimization problem which can be solved by many optimization algorithms. The complexity of three-dimensional (3D) path planning for autonomous underwater vehicles (AUVs) requires the optimization algorithm to have a quick convergence speed. This work provides a new 3D path planning method for AUV using a modified firefly algorithm. In order to solve the problem of slow convergence of the basic firefly algorithm, an improved method was proposed. In the modified firefly algorithm, the parameters of the algorithm and the random movement steps can be adjusted according to the operating process. At the same time, an autonomous flight strategy is introduced to avoid instances of invalid flight. An excluding operator was used to improve the effect of obstacle avoidance, and a contracting operator was used to enhance the convergence speed and the smoothness of the path. The performance of the modified firefly algorithm and the effectiveness of the 3D path planning method were proved through a varied set of experiments.

1. Introduction

AUVs are in strong demand within both military and civil fields. More and more research institutions are carrying out research into AUVs. Path planning is one of the key technologies in autonomous decision-making for AUVs, and it has become one of the most talked about issues in the AUV field. Many methods for AUV path planning have been put forward and most of the AUV operating environments were assumed to be two-dimensional. For example, some researchers proposed a path planning method using artificial potential field methods [1, 2]. Chen et al. brought forward a global path planning method for AUV based on the sparse A^* search algorithm [3]. Some novel optimization algorithms, such as the genetic algorithm (GA) and ant colony optimization (ACO), were also used to solve the problems of AUV path optimization [4, 5]. Although these methods were proved effective in solving path planning problems, they inevitably face some problems in practical applications because the AUV working environment is a 3D marine space.

3D path planning is a classical optimization problem. With the development of optimization algorithms, many

researchers apply new optimization algorithms to solve the path planning problem in recent years. Swarm intelligence (SI) belongs to an artificial intelligence (AI) discipline that became increasingly popular over the last decade, and it refers to a research field that is concerned with a collective behavior within self-organized and decentralized systems. Nowadays, swarm intelligence has formed one of the hottest topics in the developments of new algorithms inspired by nature. And methods of swarm intelligence are used to solve traditional NP problems and their excellent performance and great potential have been proved in many practical applications [6]. In recent years, SI-based algorithms are also applied for solving 3D path planning problem. For example, a 3D path planning method based on the ACO was proposed by Liu et al. [7, 8]. Particle swarm optimization (PSO) was also used to solve the problem of 3D path optimization [9–11]. These works prove that it is feasible to solve 3D path planning problems using SI-based algorithm.

Firefly algorithm (FA) is a more promising SI-based optimization algorithm based on firefly social features put forward by Dr. Yang in 2008. Although this algorithm is similar

to other bioinspired algorithms, it is relatively simple both in theory and in implementation. One of its most prominent advantages is that global communications can run based on individual exercise at the same time as random motion [12]. The existing research results showed that the algorithm was very effective in dealing with a lot of optimization problems, and it had gradually been applied to some kinds of optimization fields with good results [13–21]. As a new optimization algorithm, the basic FA has some drawbacks in convergence speed and stability. Some improvements have been put forward recently. For example, Dr. Yang added Levy flight to the random part of the FA formulation and Levy-flight firefly algorithm (LFA) was constructed. Experimental result showed that the improved algorithm was more effective in searching for global best values [22, 23]. Farahani et al. made some improvements to the random motion part of FA to ensure the random step is larger in the initial period, in order to avoid falling into local optima. The random step was reduced after several iterations to make the algorithm converge to the global minimum rapidly. Meanwhile, the movement style of FA was improved and directed movement was proposed. Experimental results showed that the improved FA had better performance than the basic FA [24]. dos Santos Coelho and de Andrade Bernert used chaotic sequences to adjust the parameters of FA, and the improved algorithm had better results in optimizing benchmark test functions of reliability and redundancy allocation [25]. Liu et al. made some improvements to ensure parameters of FA adjusted according to the iteration numbers. Compared to the basic FA, the performance and convergence speed were improved [26]. Farahani et al. put forward a new FA that increases convergence speed using Gaussian distribution to move all fireflies to global best in each iteration. The modified algorithm was tested on five standard functions. And experimental results showed that the modified FA had better performance than the basic firefly algorithm [27]. Fister et al. proposed a comprehensive review of the FA. The status of applications within various application areas was summarized. At the same time, the review showed that FA was simple, flexible, and versatile, which was very efficient in solving a wide range of diverse real-world problems [20].

In this paper, a modified FA and a 3D path planning method based on the new algorithm are proposed. The length of random steps and parameters of the algorithm are designed to be adjusted according to the distance between two fireflies and the iteration times. The autonomous flight strategy has been introduced to avoid invalid flight. An excluding operator and a contracting operator have been used to improve the effect of obstacle avoidance, the speed of convergence, and the smoothness of the path. The paper is organized as follows. Section 2 analyzes the performance of basic FA. Section 3 develops a modified FA. A 3D path planning method is proposed in Section 4. Experiments in Section 5 prove the performance of the modified FA and the path planning method. Finally, Section 6 contains the main conclusions and describes some problems that warrant further research.

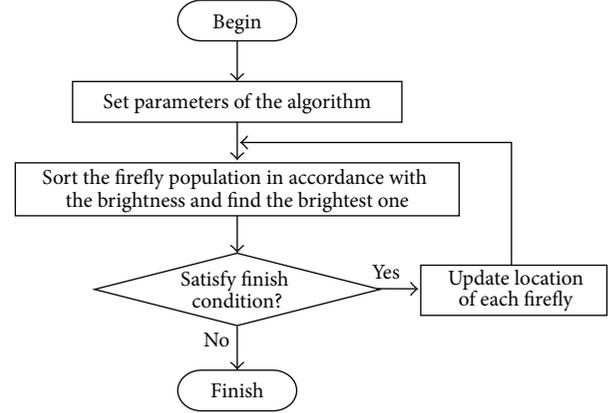


FIGURE 1: The execution process of FA.

2. Basic Firefly Algorithm and Its Performance Analysis

The FA is derived from the simplification and simulation of firefly group behavior. It has three idealized constraints which are derived from firefly features. (1) The firefly is not gender-specific. And they will fly to more attractive and larger brightness companion regardless of its gender. (2) Firefly attractive size is proportional to its brightness. And its brightness decreases with the distance between individuals. If there is no brighter or more attractive one, then it will flight randomly. (3) The brightness or attractiveness of a firefly is determined by the specified value of the objective function [12]. For the maximization problem, the luminous intensity is proportional to the value of the objective function. Based on the above three rules, the execution process of firefly algorithm is shown in Figure 1.

The core of FA is that the absolute brightness of fireflies represents the value of objective function and the position of the fireflies represents the solution of the problem to be solved. The relative brightness is gained by comparing two fireflies and it is associated with attraction. One firefly is attracted by a brighter one and adjusts its position according to this attraction. Only the brightest firefly moves randomly. The formula of FA is as follows:

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \beta_{ij}(r_{ij})(\vec{x}_j(t) - \vec{x}_i(t)) + \alpha\vec{e}_i, \quad (1)$$

$$\beta_{ij}(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2}, \quad (2)$$

where β_0 stands for the biggest attraction and γ is the absorption coefficient which controls the change in light intensity and determines the speed of convergence. α is the random coefficient which controls the range of movement. γ_{ij} is the artesian distance between firefly i and firefly j . \vec{e}_i is the random vector decided by Gauss distribution [12].

As described in formula (1), the changing position of a firefly depends on its attraction to brighter fireflies and their random movement. At the initial period of algorithm running, fireflies are randomly distributed within the search area. The distance between each of them is greater, while $\beta_{ij}(r_{ij})$ is small and the range of movement is small, so

the exploring ability of FA is insufficient with slower convergence. At the later stage of the algorithm running, fireflies gather around the optimal insect, the distance between each other is smaller, and $\beta_{ij}(r_{ij})$ is bigger. At the same time, random motion is still taking place which is not useful for the convergence of the algorithm [13].

In addition, the position updating of one firefly is affected by other brighter fireflies, some of which may not be necessarily useful for solving the problem, and this may make the new location of the firefly far away from the optimal solution (we call this behavior invalid flight). Although invalid flight may be improved by the influence of other fireflies in the following movements, it will slow down the search process and reduce the speed of convergence. Thus, in order to improve the ability of basic FA to solve practical problems, some changes are necessary.

3. A Modified Firefly Algorithm

There are two important parameters in the formula for the position updating: the absorption coefficient γ and the random coefficient α . The smaller the γ , the greater the attraction between two fireflies and the faster the convergence speed. The range of random movement is larger and the convergence speed is slower, with a bigger α . In practical optimization problems, generally global search helps the algorithm converge to an area quickly and then a local search helps obtain a high precision solution. Reference [26] proposed an adaptive FA in which α and γ were adjusted with the iteration times, and its performance was better than that of the basic FA. Because the random movement step is 1 and the changing range of the random part is -0.5 to 0.5 , the convergence speed is very slow in solving problems with a large range. In order to solve this problem, we proposed a new FA in which the random movement step was designed to be the distance between two fireflies. The position updating formula of the new FA is as follows:

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \beta_{ij}(r_{ij}) \cdot (\vec{x}_i(t) - \vec{x}_j(t)) + \alpha(t) \cdot r_{ij} \cdot \vec{\epsilon}_i; \quad (3)$$

$\beta_{ij}(r_{ij})$, $\alpha(t)$, and $\gamma(t)$ can be obtained as follows:

$$\begin{aligned} \beta_{ij}(r_{ij}) &= \beta \cdot e^{-\gamma(t)r_{ij}^2}, \\ \gamma(t) &= \gamma_b + \frac{t}{N} \cdot (\gamma_e - \gamma_b), \\ \alpha(t) &= \alpha_b + \frac{t}{N} \cdot (\alpha_e - \alpha_b), \end{aligned} \quad (4)$$

where \vec{x}_i and \vec{x}_j stand for the spatial positions of fireflies i and j ; N is the whole iteration time; $\gamma_e > \gamma_b$, where γ_b and γ_e are the initial and ultimate value of γ , respectively; $\alpha_e < \alpha_b$, where α_b and α_e are the initial and ultimate value of α , respectively.

As seen in formula (3), the attraction between two fireflies is small when the distance between them is great. Attraction therefore does not play an important role in position updating. However, as the random movement step is big, fireflies move within a big range to explore a larger search

space. When the distance among fireflies is small, the random movement step reduces and the random movement range is small. Attraction therefore begins to play a more important role in position updating. Fireflies move to the brightest one quickly and the algorithm comes to convergence at a great speed.

In addition, in order to avoid invalid flight, we introduced the autonomous flight strategy (AFS). The new position of a firefly should be judged before it moves. The firefly will move and update the position if the new position is useful for solving the problem. Otherwise, the firefly will stay at the current position.

4. 3D Path Planning for AUV Using the Modified Firefly Algorithm

4.1. Coding of FA. In order to use FA to solve 3D path planning problem, the coding of FA should be done first. If a path has n route points, it must be constituted by $n - 1$ path segments. We assume that each firefly represents a candidate path in the firefly group, and the dimensions of position for each firefly correspond to the route points. So the number of fireflies corresponds to the candidate path number. So if a firefly group contains m members and each firefly in the group has n dimensions, the (i) th candidate path $\text{Path}(i)$ represented by the (i) th firefly can be described as follows:

$$\text{Path}(i) = \{P_{i,j} \mid j = 1, \dots, n\}, \quad (5)$$

where $P_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$ stands for the (j) th route point and it corresponds to the (j) th dimension of the position for the (i) th firefly.

In addition, as the brightness represents the quality of a firefly, it corresponds to the path quality in the path planning problem. The brightest firefly represents the optimal path.

4.2. Cost Function. There are many criteria for evaluating a candidate path. Generally, the smaller the path length is, the better it is. Additionally, in order to ensure the safety of AUV, the path should be away from obstacles, so we used the path length E_L and risk value E_D to evaluate the candidate path. The cost function of path planning can be calculated as follows:

$$E = \omega_1 E_L + \omega_2 E_D, \quad (6)$$

where $\omega_1 + \omega_2 = 1$.

So the aim of path planning is to find a path which can satisfy $\min(E)$. The calculating methods of every cost part are as follows.

4.2.1. Path Length. If a path has n route points, its length can be calculated by the following formula:

$$E_L = L_{SP_1} + \sum_{i=1}^{n-1} \Delta L_i + L_{P_n E}, \quad (7)$$

where (x_i, y_i, z_i) , $1 \leq i < n$, stands for the coordinates of the (i) th route point, L_{SP_1} stands for the distance

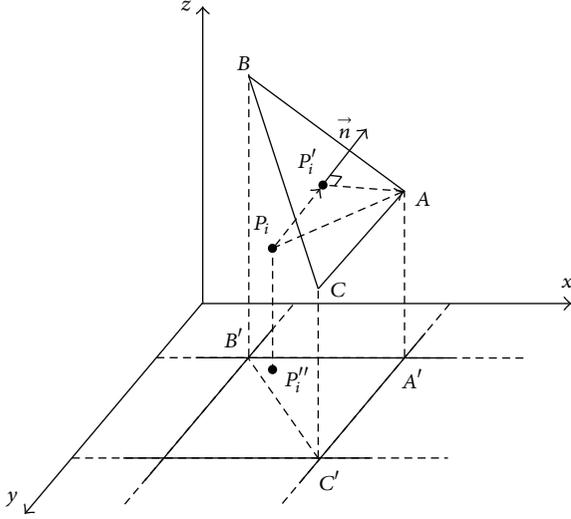


FIGURE 2: Sketch map of excluding operator.

between the start point and the first route point, $\Delta L_i = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 + (z_i - z_{i+1})^2}$ represents the distance between the (i)th route point and the ($i + 1$)th route point, and L_{P_nE} stands for the distance between the destination point and the last route point.

4.2.2. Risk Value. A path holds risks if it passes through obstacles. So the risk value can be judged by how much of a path contains obstacles. In order to calculate the risk value easily, we selected m evaluating points in every path segment. The whole risk value can therefore be judged by calculating how many route points and evaluating points which are in obstacles. For example, if there are k_1 route points and k_2 evaluating points in obstacles, the risk value can be calculated as follows:

$$E_D = C(k_1 + k_2), \quad (8)$$

where $C > 0$ stands for the cost coefficient.

4.3. Optimization Operator

4.3.1. Excluding Operator. As the path planning space is 3D and contains many obstacles, it is inevitable that some points on the path will come into contact with obstacles in the process of searching. Sometimes these points would come out of the obstacles at last, but it will cost much time. In order to reduce searching time and increase the success rate of obstacle avoidance, it is necessary to help these points avoid obstacles quickly. An excluding operator is used here to make this easier. A path point inside an obstacle will be excluded to the surface of obstacle's plane [11]. Figure 2 shows how the excluding operator works.

In Figure 2, ABC is a plane on the surface of one obstacle and \vec{n} stands for the normal direction of ABC . P_i is a path point inside the obstacle. In order to ensure safety, P_i should move out of the obstacle using an excluding operator.

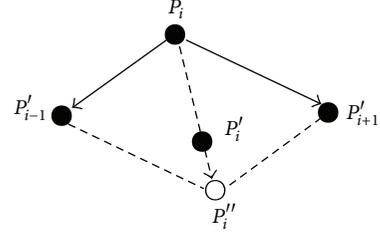


FIGURE 3: Sketch map of contracting operator.

P'_i is on plane ABC which stands for the new position of P_i after it is excluded from obstacles. The position of P'_i and the displacement of P_i can be calculated using the following formula:

$$P'_i = P_i + \vec{P_iP'_i}, \quad (9)$$

$$\vec{P_iP'_i} = \left(\vec{P_iP'_{i-1}} + \vec{P_iP'_{i+1}} \right) \vec{n}.$$

4.3.2. Contracting Operator. The generation of a safe path should be done through the cooperation of every path point. Without taking path knowledge into account, the planning process will take a long time, especially in path planning with many dimensions. A path generated by FA only may be unsmooth, so a contracting operator is used to manage the cooperation of route points and to make the path smoother [11]. Figure 3 shows how the contracting operator works.

In Figure 3, P'_{i-1} is the route point before P_i and P'_{i+1} is the route point after P_i . On the assumption that P'_{i-1} and P'_{i+1} are fixed, P_i will be pulled by these two points and the direction of the resultant force is $\vec{P_iP'_{i-1}} + \vec{P_iP'_{i+1}}$. Theoretically, with the pulling force, P_i will move to P''_i . In fact, because it is affected by air resistance, P_i can only move to P'_i . The displacement of P_i and its real position can be calculated by the following formula:

$$P'_i = P_i + \vec{S}_{P_i}, \quad (10)$$

$$\vec{S}_{P_i} = k \left(\vec{P_iP'_{i-1}} + \vec{P_iP'_{i+1}} \right),$$

where \vec{S}_{P_i} is the displacement vector of P_i with pulling forces from P'_{i-1} and P'_{i+1} and $k \in (0, 1)$ is the resistance coefficient.

The path planning flow based on the modified FA is as follows: firstly, initializing path parameters and modeling the operating environment and then searching for the optimum path using modified FA, using an excluding operator to avoid obstacles and a contracting operator to optimize the path. These steps should be repeated until a satisfactory path is found. The detailed workflow is shown in Figure 4.

5. Experimental Setup

5.1. Experiments Using Modified FA. Genetic algorithm (GA) and particle swarm optimization (PSO) are two classic

TABLE 1: Test functions.

Function	Sphere	Rosenbrock	Rastrigin	Griewank	Ackley	Zakharo
Range	[-5.12, 5.12]	[-2.048, 2.048]	[-5.12, 5.12]	[-8, 8]	[-2.768, 2.768]	[-5, 10]

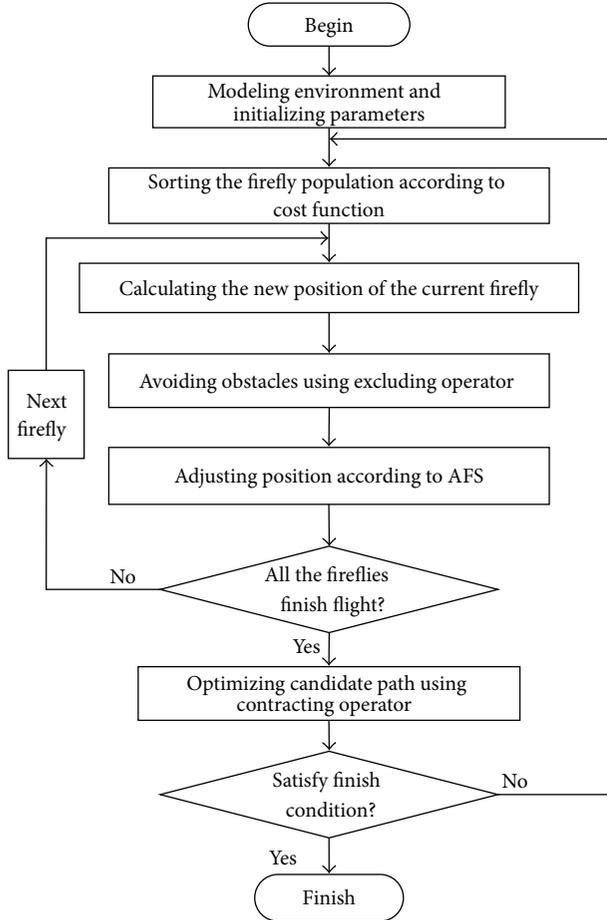


FIGURE 4: The flowchart for 3D path planning.

algorithms with extreme influence in optimization field, and research on them is relatively mature. FA has some qualities in common with these two algorithms. In order to verify the performance of the modified FA (AMFA) proposed in this paper, we used four algorithms (basic FA, AMFA, PSO, and GA) to solve six classic test functions. The four classic test functions are as follows.

Sphere: $f_1(\vec{x}) = \sum_{i=1}^n x_i^2$; the function has its global optimum $f^* = 0$ at $s^* = (0, \dots, 0)$.

Rosenbrock: $f_2(\vec{x}) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$; the function has global optimum $f^* = 0$ at $s^* = (1, \dots, 1)$.

Rastrigin: $f_3(\vec{x}) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$; the function has global optimum $f^* = 0$ at $s^* = (0, \dots, 0)$.

Griewank: $f_4(\vec{x}) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$; the function has global optimum $f^* = 0$ at $s^* = (0, \dots, 0)$.

Ackley: $f_5 = -20 \exp[-0.2 \sqrt{(1/d) \sum_{i=1}^d x_i^2}] - \exp[(1/d) \sum_{i=1}^d \cos(2\pi x_i)] + (20 + e)$; the function has global optimum $f^* = 0$ at $s^* = (0, \dots, 0)$.

Zakharo: $f_6 = \sum_{i=1}^D s_i^2 + ((1/2) \sum_{i=1}^D i s_i)^2 + ((1/2) \sum_{i=1}^D i s_i)^4$; the function has global optimum $f^* = 0$ at $s^* = (0, \dots, 0)$.

Parameters of these four algorithms are set as follows.

FA: $\beta_0 = 1.0$, $\gamma = 1.0$, and $\alpha = 1.0$. rand was the uniformly distributed random number in the range of 0-1.

GA: binary encoding was used and the binary digit was 20. The roulette wheel selection method and single point crossover were used. Crossover probability was defined as $p_c = 0.6$, mutation probability was described as $p_m = 1/n$, and n was the dimension of the optimization function.

PSO: using parameters recommended by Li et al. [10], $\omega = 0.729$ and $c_1 = c_2 = 1.494$. The maximum speed was set to the top of the search range, $V_{\max} = X_{\max}$.

AMFA: $\beta_0 = 1.0$, $\gamma_e = 2.0$, $\gamma_b = 1.0$, $\alpha_e = 0.8$, and $\alpha_b = 2.0$. rand was the uniformly distributed random number in the range of 0-1.

In order to compare the optimization performance of these algorithms fairly and sufficiently, they were used to solve the four test functions in both small and wide ranges. The population size of every algorithm was set to 40 and the dimension of the testing function was set to 30. Each algorithm was run 30 times independently, and the number of fitness evaluations was limited to 60,000.

5.1.1. *Experiments in a Small Range.* The range of every test function is shown in Table 1. Table 2 shows the results of simulation for every algorithm. In order to observe the iterative process of the algorithm, the curves of the six test functions are shown in Figures 5 to 10.

Experimental results in Table 2 show that the precision of solution obtained by AMFA is much better than that of FA. Meanwhile, Figures 5–10 show that the convergence speed of AMFA is quicker than that of FA. So we can conclude that the performance of AMFA is better than that of FA and the method proposed here is effective in improving the performance of basic FA. The performance and convergence speed of AMFA are better than those of PSO and GA. Although the solution precision of Sphere obtained by AMFA is not the best, it is valid. Meanwhile, the solution of Sphere

TABLE 2: Experimental results in a small range.

Function	Term	AMFA	FA	PSO	GA
Sphere	Best	0	$5.20E - 09$	$5.05E - 41$	$9.64E - 07$
	Mean	$5.58E - 13$	$2.76E - 06$	$1.75E - 36$	$4.91E - 06$
	STD	$1.73E - 12$	$3.58E - 06$	$9.19E - 36$	$4.56E - 06$
	"0" times	22	0	0	0
Rosenbrock	Best	0	$3.66E - 07$	$1.27E - 02$	7.8507
	Mean	$2.38E - 06$	$6.70E - 05$	$2.17E - 01$	51.3280
	STD	$4.34E - 07$	$1.07E - 04$	0.75	29.7729
	"0" times	21	0	0	0
Rastrigin	Best	$00E + 00$	$1.98E - 07$	1.00	43.78
	Mean	$00E + 00$	$9.81E - 06$	4.88	58.04
	STD	$00E + 00$	$1.70E - 04$	2.24	8.28
	"0" times	30	0	0	0
Griewank	Best	$00E + 00$	$1.35E - 10$	$00E + 00$	$1.14E - 07$
	Mean	$3.20E - 11$	$5.36E - 07$	$5.6E - 03$	$2.30E - 02$
	STD	$1.12E - 10$	$8.51E - 07$	$8.3E - 03$	$4.40E - 02$
	"0" times	26	0	7	0
Ackley	Best	0	$3.16E - 05$	$1.42E - 14$	$1.34E + 00$
	Mean	$1.75E - 07$	$3.12E - 04$	$3.02E - 01$	$2.34E + 00$
	STD	$7.23E - 07$	$2.69E - 04$	$5.68E - 01$	$3.71E - 01$
	"0" times	21	0	0	0
Zakharo	Best	$00E + 00$	$1.7E - 07$	$4.6E - 06$	$9.8E + 01$
	Mean	$1.6E - 06$	$2.4E - 03$	$6.7E - 05$	$1.8E + 02$
	STD	$9.0E - 06$	$3.8E - 03$	$6.8E - 05$	$4.6E + 01$
	"0" times	4	0	0	0

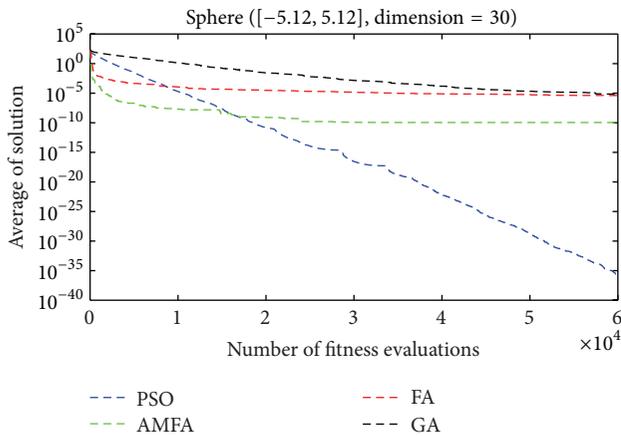


FIGURE 5: Test results of Sphere.

obtained by AMFA is found to be zero 22 times. We can therefore conclude that AMFA is an effective algorithm for solving problems with a small range.

5.1.2. Experiments in a Wide Range. The range of every test function is shown in Table 3. Table 4 shows the results of simulation for every algorithm. In order to observe the iterative process of the algorithm, the curves of the six test functions are shown in Figures 11–16.

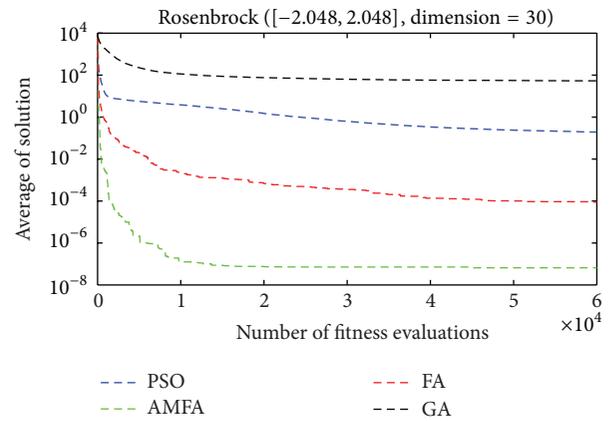


FIGURE 6: Test results of Rosenbrock.

Experimental results in Table 4 show that the precision of solution obtained by AMFA is the most effective for solving all six test functions. Additionally, it can find the global optimum every time. Figures 11 to 16 show that the convergence speed of AMFA is the quickest of the four algorithms. So we can conclude that the method proposed here is effective in improving the performance of basic FA and AMFA is powerful in solving problems with a wide range.

5.2. Experiments of Path Planning. In order to simulate the real sailing environmental space of AUV, a maritime space

TABLE 3: Test functions.

Function	Sphere	Rosenbrock	Rastrigin	Griewank	Ackley	Zakharo
Range	$[-100, 100]$	$[-100, 100]$	$[-100, 100]$	$[-600, 600]$	$[-23.768, 23.768]$	$[-50, 100]$

TABLE 4: Experimental results in wide range.

Function	Term	AMFA	FA	PSO	GA
Sphere	Best	00E + 00	5.87E - 09	5.23E - 39	3.42E - 04
	Mean	00E + 00	4.00E - 03	8.73E - 35	1.71E - 03
	STD	00E + 00	2.15E - 02	2.16E - 34	1.52E - 03
	"0" times	30	0	0	0
Rosenbrock	Best	00E + 00	5.95E - 07	2.58E - 02	2.88E + 00
	Mean	00E + 00	6.00E + 04	8.61E - 01	7.67E + 01
	STD	00E + 00	3.28E + 05	1.46E + 00	4.98E + 01
	"0" times	30	0	0	0
Rastrigin	Best	00E + 00	6.11E - 05	2.98E + 00	1.99E + 01
	Mean	00E + 00	8.13E + 01	6.40E + 00	6.08E + 01
	STD	00E + 00	2.54E + 02	2.38E + 00	3.15E + 01
	"0" times	30	0	0	0
Griewank	Best	00E + 00	4.05E + 02	00E + 00	4.85E - 04
	Mean	00E + 00	5.72E + 02	2.30E - 02	5.45E - 02
	STD	00E + 00	5.52E + 01	2.85E - 02	5.30E - 02
	"0" times	30	0	2	0
Ackley	Best	0	3.57E - 05	2.13E - 14	4.10E - 03
	Mean	00E + 00	2.30E - 03	1.63E + 00	7.7E - 03
	STD	0	4.40E - 03	1.12E + 00	3.1E - 03
	"0" times	30	0	0	0
Zakharo	Best	0	2.0E - 06	4.8E + 01	1.5E + 04
	Mean	1.5E - 322	1.5E - 01	3.3E + 02	3.0E + 04
	STD	0	7.6E - 01	2.8E + 02	6.2E + 03
	"0" times	5	0	0	0

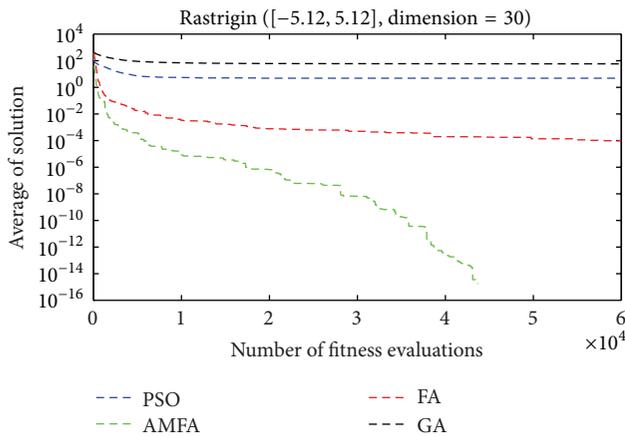


FIGURE 7: Test results of Rastrigin.

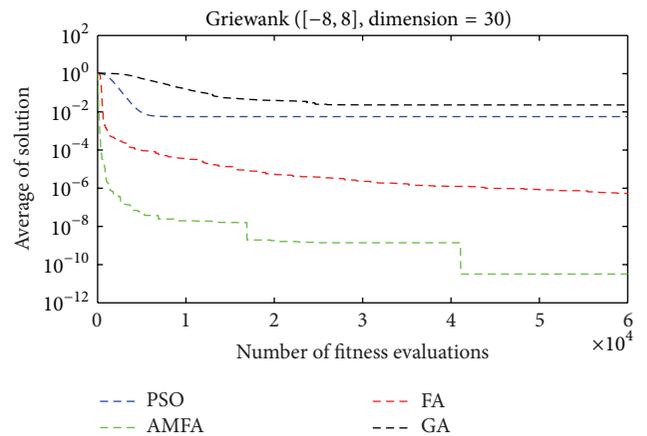


FIGURE 8: Test results of Griewank.

with a horizontal area about $2^\circ \times 2^\circ$ was selected as the experiment environment for AUV path planning. The terrain was generated by interpolation with elevation data extracted from electric chart. In order to observe the obstacle avoidance

capability of the method, the actual lengths of the horizontal and vertical coordinates were reduced $(1852 \times 60)/1000$ times. The resistance coefficient was 0.9553 and parameters of AMFA were the same as in the experiments described above. Because the population size of AMFA was 40 and

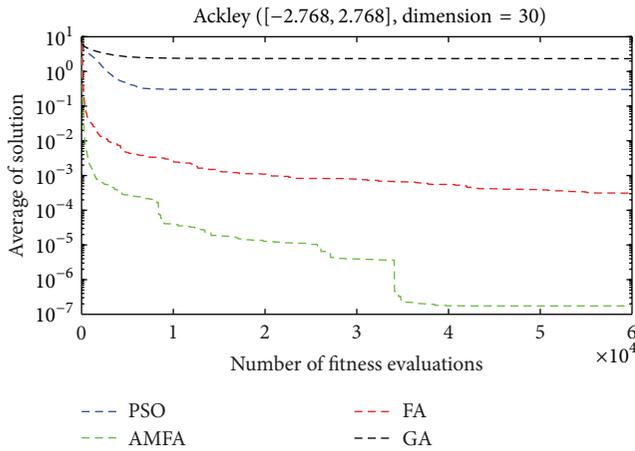


FIGURE 9: Test results of Ackley.

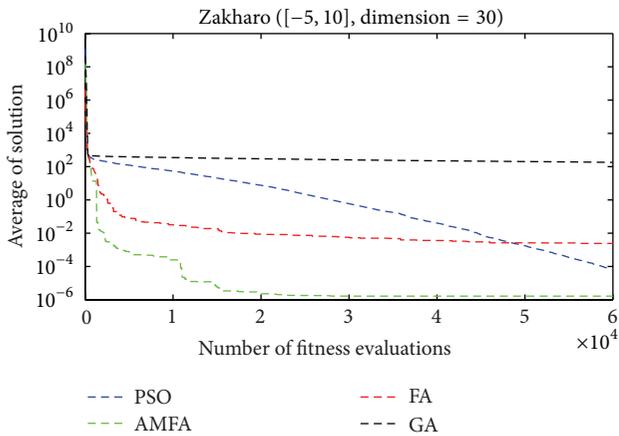


FIGURE 10: Test results of Zakharo.

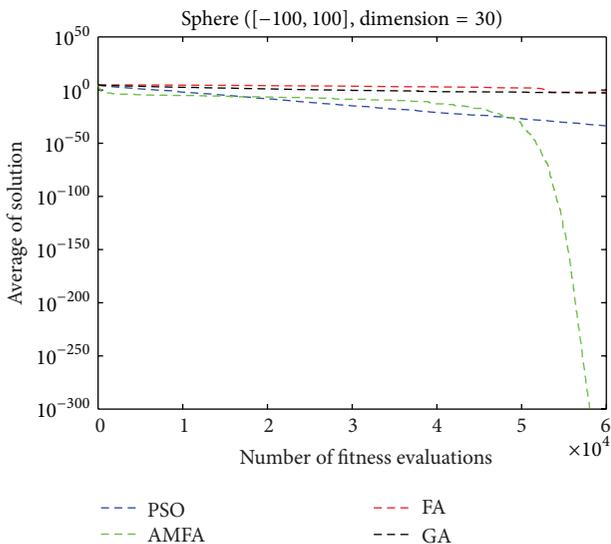


FIGURE 11: Test results of Sphere.

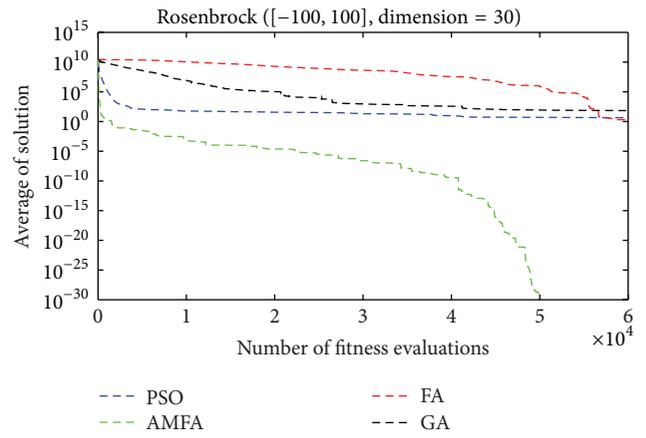


FIGURE 12: Test results of Rosenbrock.

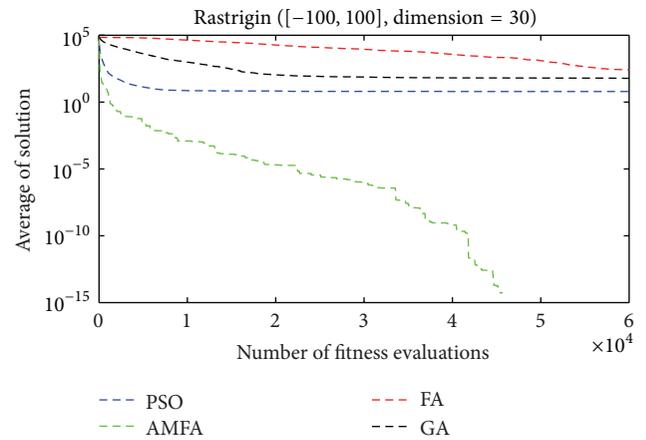


FIGURE 13: Test results of Rastrigin.

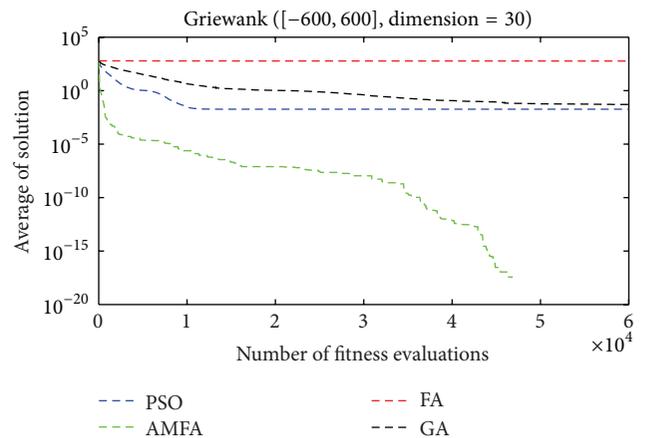


FIGURE 14: Test results of Griewank.

the dimension of every firefly was set to 30, the number of candidate paths was 40 and every path should have 30 path points.

In order to verify the performance of the new algorithm, two experiments were designed here. In the first experiment, depths of the starting point and destination

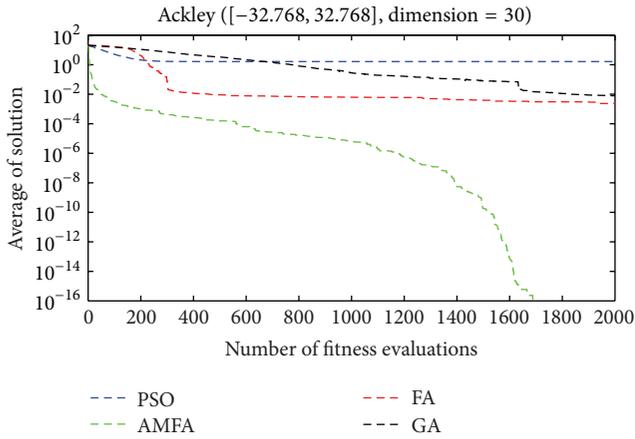


FIGURE 15: Test results of Ackley.

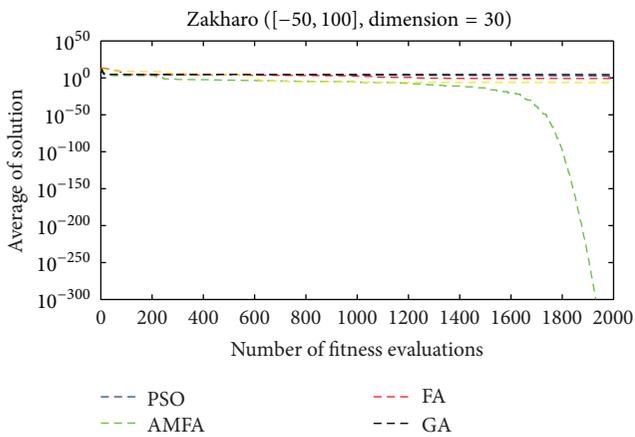


FIGURE 16: Test results of Zakharo.

are both 300 meters. The horizontal position of starting point is (111.10°E, 17.07°N), and the horizontal position of destination is (112.50°E, 16.69°N). The results of the path planning are shown in Figure 17. In the second experiment, the start position and destination have different depths. The horizontal position of the starting point is (111.30°E, 15.87°N) and the depth is 230 meters. The horizontal position of the destination is (112.80°E, 17.11°N), and the depth is 450 meters. The results of the path planning are shown in Figure 18.

Figures 17 and 18 show that there are many obstacles in the environmental space. The line represents the path found by AMFA, the black dot is the starting point, and the red dot stands for the destination in the figures. From the results of the experiments we can conclude that the path planning method using AMFA can find the optimal path in the complex environmental space, and the paths in both experiments are smooth and do not path through obstacles. The experimental results prove that the path planning method proposed here is feasible and AMFA can be used in solving 3D path planning problem.

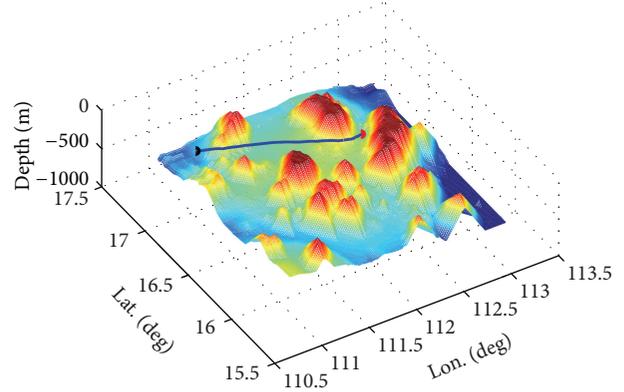


FIGURE 17: Results of experiment 1.

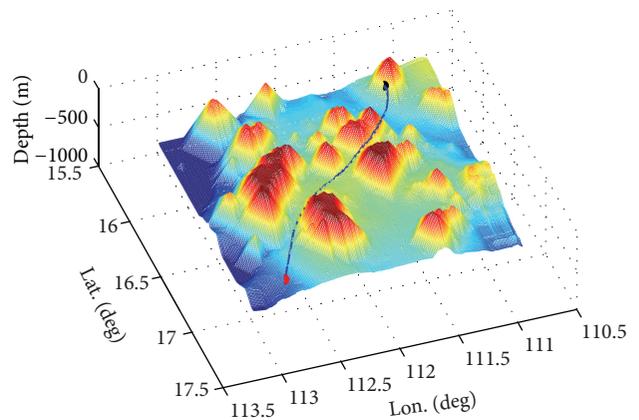


FIGURE 18: Results of experiment 2.

6. Conclusions

This paper presents a 3D path planning method based on a modified FA. The performance of the modified FA and the path planning method was verified through simulation. The simulation results show that the modified FA has a quick convergence speed and the path planning method based on it can find an effective path in a 3D environment. However, as a new SI-based algorithm, the firefly algorithm still has imperfections from a theoretical research point of view and its mathematical theory also needs to be studied and proved. Additionally, it has been assumed that the operating environment is static and obstacle information is known before the path planning method is begun. In reality, the marine environment is dynamic and AUVs will face many sudden threats while in action. Finally, many constraints for AUVs such as the maximal turning angle and the maximal sailing depth must be taken into account. Therefore further research is required to conclude how to plan a practical path for AUVs in the real dynamic marine environment.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China under Grant no. 51379049 and Foundation of Central University HEUCFX41302 and HEUCF041416.

References

- [1] F. Wang, L. Wan, Y. R. Xu, and Y. K. Zhang, "Path planning based on improved artificial potential field for autonomous underwater vehicles," *Journal of Huazhong University of Science and Technology*, vol. 39, no. 11, pp. 184–187, 2011.
- [2] X. Li and D. Q. Zhu, "Path planning for autonomous underwater vehicle based on artificial potential field method," *Journal of Shanghai Maritime University*, vol. 31, no. 2, pp. 35–39, 2010.
- [3] S. Chen, C. W. Liu, Z. P. Huang, and G. C. Cai, "Global path planning for AUV based on sparse A* search algorithm," *Torpedo Technology*, vol. 20, no. 4, pp. 271–275, 2012.
- [4] G. Yan, L. Wang, J. Zhou, and J. Cha, "Path planning based on improved genetic algorithm for AUV," *Journal of Chongqing University of Technology (Natural Science)*, vol. 24, no. 5, pp. 81–85, 2010.
- [5] J. H. Wang, H. X. Wu, and X. C. Shi, "Global path planning for AUV based on ACO," *Shipbuilding of China*, vol. 49, no. 2, pp. 88–93, 2008.
- [6] I. Fister Jr., X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," *Elektrotehniški Vestnik*, vol. 80, no. 3, pp. 1–7, 2013.
- [7] L. Q. Liu, Y. T. Dai, L. H. Wang, and X. L. Gan, "Research on global path planning of underwater vehicle based on ant colony algorithm," *Journal of System Simulation*, vol. 19, no. 18, pp. 4174–4177, 2007.
- [8] L.-Q. Liu, F. Yu, and Y.-T. Dai, "Path planning of underwater vehicle in 3D space based on ant colony algorithm," *Journal of System Simulation*, vol. 20, no. 14, pp. 3712–3716, 2008.
- [9] J. Sun and S. T. Wu, "Route planning of cruise missile based on improved particle swarm algorithm," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 37, no. 10, pp. 1228–1232, 2011.
- [10] M. Li, D. B. Wang, T. T. Bai, and S. Z. Sheng, "Route planning based on particle swarm optimization with threat heuristic," *Electronics Optics & Control*, vol. 18, no. 12, pp. 1–4, 2011.
- [11] Z. C. Fan, *Path planning method based on the algorithm of PSO and elastic rope for underwater vehicle in three-dimensional space [Master thesis]*, Harbin Engineering University, 2013.
- [12] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, 2008.
- [13] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications*, vol. 5792 of *Lecture Notes in Computer Sciences*, pp. 169–178, Springer, Berlin, Germany, 2009.
- [14] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimization," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010.
- [15] R. Dutta, R. Ganguli, and V. Mani, "Exploring isospectral spring-mass systems with firefly algorithm," *Proceedings of The Royal Society of London Series A: Mathematical, Physical and Engineering Sciences*, vol. 467, no. 2135, pp. 3222–3240, 2011.
- [16] T. Mauder, C. Sandera, J. Stetina, and M. Seda, "Optimization of the quality of continuously cast steel slabs using the firefly algorithm," *Materials and Technology*, vol. 45, no. 4, pp. 347–350, 2011.
- [17] P. Aungkulanon, P. Chai-ead, and P. Luangpaiboon, "Simulated manufacturing process improvement via particle swarm optimisation and firefly algorithms," in *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS '11)*, pp. 1123–1128, March 2011.
- [18] T. Apostolopoulos and A. Vlachos, "Application of the firefly algorithm for solving the economic emissions load dispatch problem," *International Journal of Combinatorics*, vol. 2011, Article ID 523806, 23 pages, 2011.
- [19] M.-H. Horng and T.-W. Jiang, "Multilevel image thresholding selection based on the firefly algorithm," in *Proceedings of the 7th International Conference on Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC '10)*, pp. 58–63, October 2010.
- [20] I. Fister, I. Fister Jr., X.-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms," *Swarm and Evolutionary Computation*, vol. 13, pp. 34–46, 2013.
- [21] I. Fister, X.-S. Yang, D. Fister, and I. Fister Jr., "Firefly algorithm: a brief review of the expanding literature," in *Cuckoo Search and Firefly Algorithm*, vol. 516 of *Studies in Computational Intelligence*, pp. 347–360, 2014.
- [22] X.-S. Yang, "Firefly algorithm, Lévy flights and global optimization," in *Research and Development in Intelligent Systems XXVI*, pp. 209–218, Springer, London, UK, 2010.
- [23] X.-S. Yang and S. Deb, "Eagle strategy using Levy walk and firefly algorithms for stochastic optimization," in *Nature Inspired Cooperative Strategies for Optimization*, vol. 284, pp. 101–111, Springer, Berlin, Germany, 2010.
- [24] S. M. Farahani, A. A. Abshouri, B. Nasiri, and M. Meybodi, "An improved firefly algorithm with directed movement," in *Proceedings of 4th IEEE International Conference on Computer Science and Information Technology*, pp. 248–251, 2011.
- [25] L. dos Santos Coelho and D. L. de Andrade Bernert, "A chaotic firefly algorithm applied to reliability-redundancy optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 517–521, June 2011.
- [26] C. Liu, Z. Gao, and W. Zhao, "A new path planning method based on firefly algorithm," in *Proceedings of the 5th International Joint Conference on Computational Sciences and Optimization (CSO '12)*, pp. 775–778, June 2012.
- [27] S. M. Farahani, A. Abshouri, B. Nasiri, and M. Meybodi, "A gaussian firefly algorithm," *International Journal of Machine Learning and Computing*, vol. 1, no. 5, pp. 448–454, 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

