International Journal of Computational Intelligence and Applications

A NEW FUZZY FIREFLY ALGORITHM WITH ADAPTIVE PARAMETERS

TAHEREH HASSANZADEH*

Department of IT and Computer Engineering, Qazvin Branch, Islamic Azad University, Qazvin, IRAN
t.hassanzadeh@qiau.ac.ir

MOHAMMAD REZA MEYBODI

Department of IT and Computer Engineering, Amirkabir University of Technology, Tehran, IRAN
mmeybodi@aut.ac.ir

MASOUMEH SHAHRAMI RAD

University of Applied science and Technology, Center of Mehr Pirooz, Mazandaran, IRAN
shahramirad.m@gmail.com

Received Day Month Day
Revised Day Month Day

Firefly algorithm is a swarm based algorithm that can be used for solving optimization problems. This paper proposed an improved Fuzzy Adaptive Firefly Algorithm (FAFA). In the proposed FAFA, a fuzzy system is used to adapt Firefly Algorithm's parameters in order to improve its ability in global and local searches. Also, we used different fireflies initializing intervals and different iteration numbers to show the algorithm capability to find global optima. Results focus on the two case study categories of function optimization (seven benchmark functions) and presented a novel optimal multilevel thresholding approach for histogram-based image segmentation by using proposed FAFA and Otsu method. Evidence indicates that the optimization results of proposed FAFA approach are so better than the standard FA.

Keywords: Firefly Algorithm; Fuzzy Adaptive Firefly Algorithm; Multilevel Tresholding; Image Segmentation; Fuzzy System.

1. Introduction

Collective artificial intelligence is a type of intelligence based on the collective behavior of agents in decentralized and self-organized systems. These systems are usually organized from simple agents that cooperate locally with other elements and environment. Consequently, nature inspired metaheuristic algorithms are becoming more powerful algorithms for solving optimization problems[1-5].

There are different methods for optimization. Particle Swarm Optimization (PSO)[6], Ant Colony Optimization (ACO)[7], Artificial Fish Swarm Algorithm (AFSA)[8] and Artificial Bee Colony (ABC)[9] are the most well-known algorithms have ever been proposed for optimization. Idealizing some of the flashing characteristics of the fireflies,

* Address correspondence to Tahereh Hassanzadeh, Department of IT and Computer Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran. E-mail: t.hassanzadeh@qiau.ac.ir.

Firefly Algorithm was introduced by Xin-She Yang (2008)[10] at Cambridge University. In this bio-inspired algorithm the elements are fireflies with specified brightness. The firefly's brightness depends on the objective function and its movement in the environment is based on the other fireflies' brightness. Yang (2010)[11] also introduced a new version of FA (Levy FA) which combined Levy flight with the search strategy via the firefly to improve the randomization of FA. Discrete Firefly Algorithm has been also introduced by Sayadi et al. (2010)[12]. Also this algorithm is applied in various fields of image processing application. In Ref. 13, firefly algorithm used for designing a codebook for image segmentation. Also, FA applied for vector quantization[14]. In addition, FA and minimum cross entropy algorithm used for image segmentation[15]. Moreover, In Ref. 16, FA applied FA for multilevel thresholding of electrophoresis images.

There are two important parameters in the standard Firefly Algorithm which specify the behavior of the algorithm: alpha (α) and gamma (γ). If the values of these two parameters are large, the algorithm will be good in global search behavior and vice-versa in local search. In this paper, we proposed a new Fuzzy Adaptive Firefly Algorithm to hold a balance between local search and global search ability of the Firefly Algorithm. Then we combined this new optimal FAFA with the well-known Otsu method[17] for image segmentation. The results proved better performance of the proposed segmentation algorithm.

The rest of the paper is organized as follows: Section 2 describes the Firefly Algorithm. Section 3 gives a detailed description of the proposed FAFA. The proposed FAFA experimental results are discussed in section 4. Section 5 provides a brief detail of the Otsu method and the proposed segmentation method is discussed in Section 6. The experimental results of proposed segmentation method discussed in section 7 and Section 8 conclude the paper.

2. Firefly Algorithm

The Firefly algorithm is a kind of metaheuristic algorithm inspired by luminosity characteristic of fireflies' behaviors in the nature. In the Firefly algorithm, there are three idealized rules:

- All fireflies are unisex. Accordingly, a firefly will be attracted to other fireflies regardless of their sex;
- Attractiveness is proportional to their brightness. Thus, for any two flashing fireflies, the less bright one will move towards the brighter one. Attractiveness is proportional to brightness and they both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly;
- The brightness of a firefly is affected or determined by the landscape of the objective function. For a maximization problem, the brightness can simply be proportional to the value of the objective function[10].

The pseudo code of these three rules can be illustrated as in Fig. (1).

There are two important issues in the FA: variation of light intensity and the formulation of the attractiveness. For simplicity assume that the attractiveness of fireflies is based on their brightness and it is associated with the objective function. Light intensity decreases when the distance from the light source increases; moreover light is absorbed in the environment. Accordingly, the light intensity can be formulated as follows:

$$I = I_0 e^{-\gamma r^2} \tag{1}$$

where I_0 is light intensity of the light source, γ is light absorption coefficient and r is the distance between firefly i and j.

A firefly's attractiveness is proportional to the light intensity seen by the adjacent fireflies; therefore we can define the attractiveness of a firefly as:

$$\beta(r) = \beta_0 e^{-\gamma r^2} \tag{2}$$

where, β_0 is the attractiveness at r= 0.

It should be noted that the $r_{i,j}$ which is described by Eq. (3), is the Cartesian distance between any two fireflies i and j at x_i and x_j, where, x_i and x_j are the spatial coordinates of the fireflies i and j, respectively.

$$r_{i,j} = \left\| x_i - x_j \right\| = \sqrt{\sum_{k=1}^{d} \left(x_{i,k} - x_{j,k} \right)^2} \tag{3}$$

The movement of a Firefly i, which is attracted to another more attractive Firefly j, is determined by:

$$X_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} \left(x_j - x_i \right) + \alpha \left(rand - \frac{1}{2} \right) \tag{4}$$

where, the second term is the attraction while the third one is randomization including randomization parameter α and the random number generator *rand* with its values uniformly distributed in interval [0, 1].

For the most cases of implementations, $\beta_0 = 1$ and $\alpha \in [0, 1]$[18]. The parameter γ is light absorption coefficient and characterizes the variation of the attractiveness and its value is important to determine the speed of the convergence and how the FA behaves. In the most applications, it typically varies from 0.01 to 100 (ibid.).

Firefly algorithm
Initialize algorithm parameters:
MaxGen: the maximum number of generations
*Objective function of f(***x***), where* **x***=*$(x_1,........,x_d)T$
Generate initial population of fireflies or $\mathbf{x}_i$ *(i=1, 2,..., n)*
Define light intensity of $\mathbf{I}_i$ *at* $\mathbf{x}_i$ *via* $f(\mathbf{x}_i)$
While *(t<MaxGen)*
For *i = 1 to n (all n fireflies);*
For *j=1 to n (all n fireflies)*
If *($I_j > I_i$), move firefly i towards j;* ***end if***
Attractiveness varies with distance r via Exp [-γ^{r2}];
Evaluate new solutions and update light intensity;
End for j;
End for i;
End while;
Post process results and visualization;
End procedure;

Fig. 1: Pseudo code of the FA.

3. The Proposed Fuzzy Adaptive Algorithm

In the Firefly Algorithm, there are two important parameters including alpha (α), that can determine the length of the random walk of fireflies, and gamma (γ) that is light absorption coefficient in the air and determines the convergence speed of the fireflies. The behavior and convergence speed of the algorithm depend on these two parameters.

In the standard FA, determining the initial values of α and γ influences the final results essentially. Values of these two parameters remain constant; they equal to the initial values during algorithm execution. If greater initial value is considered for α parameter, fireflies will move with longer steps in each iteration. In this condition, fireflies are more powerful in escaping from the local optimums. But in this case there will be some problems; accuracy and consistency of the algorithm will decrease in this situation.

In fact, the algorithm acts better in global searching, but after approaching the global optimum, it would be incapable of appropriate local search because of the fact that α is larger than it should be. Therefore, owing the large value of α, positions with better fitness are unlikely to be found and fireflies will pass the global optimum, even they may go far from it. Considering smaller values for this parameter makes the algorithm more consistent and accurate, but it causes the algorithm to move towards the target more slowly and makes it incapable of escaping from local optimums.

In the Firefly Algorithm, the γ parameter also has an important role in the algorithm behavior. There are two important limiting cases when $\gamma \to 0$ and $\gamma \to \infty$. For $\gamma \to 0$, the attractiveness is constant $\beta = \beta_0$, that is, the light intensity does not decrease in idealized sky. Thus, a flashing firefly can be seen anywhere in the domain. As a result, a single (usually global) optimum can easily be reached. This corresponds to a special case of particle swarm optimization (PSO). Subsequently, the efficiency of this special case is the same as the PSO[18].

On the other hand, the case $\gamma \to \infty$ leads to $\beta(r) \to \delta(r)$ (the Dirac delta function), which means that the attractiveness is almost zero in the sight of the other fireflies or the fireflies are short-sighted. This is the same as the case where the fireflies randomly fly in a very foggy region. No other fireflies can be seen, and each firefly roams in a completely random way. Therefore, this corresponds to the completely random search method[18]. If greater initial values are considered for γ parameter, fireflies will move randomly and can explore the environment. But they may not be able to find the global optimum.

Accordingly, it may seems that the right solution to determine algorithm's parameters is to set large value for parameters and then decrease them due to the algorithm iterations. But in this case, we just use the algorithm iteration as a criterion to reduce α and γ parameters and this reduction will be based on a fixed value. In this case, there is no control over the obtained results from the algorithm; moreover, it is not checked whether this reduction should be extended or how much the reduction coefficient should be. To solve these problems, we use a fuzzy engine to control the two parameters' reduction procedure. This fuzzy engine considers the number of algorithm's iterations and success rate to set an appropriate amount for coefficient recreation rate of α and γ in each iteration.

Based on the above statements, larger initial values for α and γ are selected at first in order to obtain better results. Afterward, these values are reduced adaptively during the algorithm execution using the output of fuzzy engine. As a result, fireflies move quickly towards the target and are more capable of escaping local optimums. Then, by approaching the target, fireflies can accurately investigate the environment by smaller α and γ.

In order to control the values of α and γ and hold a balance between global search and local search, a new parameter, called *adaptive weight*, has been proposed here. Weight must be greater than 0 and smaller than 1. In the current iteration α and γ values are calculated according to the following formulas in presence of weight;

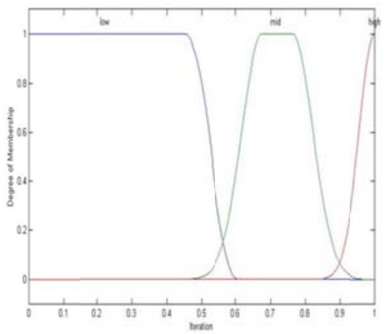
$$alpha_{inr} = w \times alpha_{itr-1} \tag{5}$$

$$gamma_{itr} = w \times gamma_{itr-1} \tag{6}$$

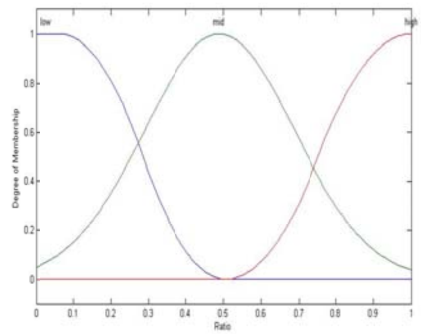
where, *W* is *adaptive Weight* which is generated as an output of the proposed fuzzy engines described in the following parts of this paper. $Alpha_{itr}$ and $gamma_{itr}$ stand for the current iterations of α and γ and $alpha_{itr-1}$and $gamma_{itr-1}$are the pervious iterations of α and γ respectively.

In this method, weight is a value between 0 and 1 that is calculated as an output of the fuzzy engine. In each iteration α and γ would be set based on the output weight. The proposed fuzzy engine has two inputs and one output: *Iteration Number* and *Ratio of Improved Fireflies* as inputs and *adaptive weight* as an output. *Iteration Number*, normalized between 0 and 1, is the proportion of current iteration number to the final iteration number. In fact, α and γ parameters must be larger in initial iterations to achieve better global searching. Therefore, α and γ values decrease very smooth in initial iterations of the algorithm execution. The proposed Progress of the algorithm causes the fireflies come close to the global optimum of the problem. Therefore, in order to increase the local search capability of the algorithm, α and γ parameters must be reduced by larger amounts. As a result, fireflies are able to search the global optimum more keenly. Considering the above facts, approaching the final iterations, the proposed fuzzy engine increases the adaptive weight to reduce α and γ more sharply.

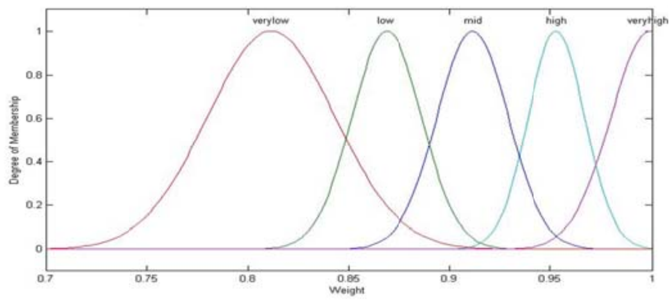
Ratio of Improved Fireflies is the proportion of the number of fireflies that find better positions in problem space (points with higher fitness) to the total number of fireflies in comparison with previous iteration. When most of the fireflies find better positions compared with previous iteration, α and γ parameters are suitable. Therefore, there is no need to reduce them. In this situation, *Ratio of Improved Fireflies* value is a number close to 1. Conversely, when most of the fireflies do not experience any improvement over the previous iteration the adaptive weight must be increased to reduce α and γ. Reducing α and γ raises the probability of finding positions with better fitness values due to the fact that it increases the local capability of the fireflies. With regard to the above issues, the proposed fuzzy engine increases the adaptive weight by reducing the ratio of the improved firefly's value and vice versa. As illustrated in Fig. (2a), we divide the number of iterations of the algorithm into three sections. Also, as shown in Fig. (2b), the rate of FA progress is divided into three parts. The weight for the next iteration is determined according to the number of FA iterations and the ratio of improved FA. The combination of these two values is performed by a certain rules as illustrated in Table (1). Adaptive Weight can be obtained with regard to the inputs of inference engine and the defined rules. The output weight will be in the range determined in Fig. (2c). These rules are defined by a certain goal. For example, if after a low iteration the rate of FA progress is very good, the weight must be near 1 because worms do not spread and must maintain their position and minimize change of the fireflies' position is necessary. The proposed fuzzy engine, which is a Mamdani fuzzy inference system with centroid of area defuzzification strategy, uses the rules illustrated in the fuzzy associative memory in Table (1). The pseudo code of the proposed FAFA is demonstrated in Fig. (3).



Iteration number (a)



Ratio of Improved Fireflies (b)



Adaptive weight (c)

Fig. 2: Fireflies Membership functions: (a) Iteration Number, (b) Ratio of Improved Fireflies, (c) Adaptive Weight.

Table 1: Fuzzy associative memory for the proposed fuzzy engine. VL: very low, L: low, M: mid, H: high and VH: very high.

Iteration number	Ratio of improved firefly	Weight
L	H	**VH**
L	M	**H**
L	L	**M**
M	H	**H**
M	M	**M**
M	L	**L**

H	H	**M**
H	M	**L**
H	L	**VL**

Proposed algorithm
Initialize algorithm parameters:
MaxGen: the maximum number of generations
Objective function of $f(\mathbf{x})$, where $\mathbf{x}=(x_1,........,x_d)T$
Generate initial population of fireflies or $\mathbf{x}_i$ (i=1, 2,..., n)
Define light intensity of $\mathbf{I}_i$ at $\mathbf{x}_i$ via $f(\mathbf{x}_i)$
While *(t<MaxGen)*
For *i = 1 to n (all n fireflies);*
For *j=1 to n (all n fireflies)*
If *($I_j > I_i$), move firefly i towards j;* ***end if***
Attractiveness varies with distance r via Exp [$-\gamma^{r^2}$];
Evaluate new solutions and update light intensity;
End for j;
End for i;
Normalize the iteration number via Iter = t/MaxGen;
Find the ratio of improvement.
Evaluate the value of α and γ by using fuzzy inference system.
End while;
Post process results and visualization;
End procedure;

Fig. 3: Pseudo code of the proposed FAFA.

4. Experimental Results of the Proposed Fuzzy Adaptive Firefly Algorithm

In this section, the aforementioned seven functions are utilized to evaluate the proposed FAFA. All of them are standard test functions. The applied functions together with their search space range are presented in Table (2). It should be mentioned that the global optimum solution of the above functions is 0. In this paper, for evaluation of the proposed algorithm, we execute the standard FA and proposed FAFA 30 times with different iteration numbers and different initializations. To evaluate the effect of the initialization of the standard firefly algorithm and proposed algorithm, we initialized the algorithms in two different intervals including; the function space and then at interval [0, 1] and the largest iteration numbers are set to 100 for the seven functions. We test the mentioned functions in 10, 20 and 30 Dimensional spaces. The population size for 10-D, 20-D and 30-D set as 50. The initial value of γ is set as 100 and the initial value of α is set to be 30 percent of the length of search space. For example, for step function the length of search space is 200 and the 30 percent of 200 is around 60 and β_0=1.

Experiments were repeated 30 times and the best, mean, worst and standard deviations which are obtained from the mentioned functions are given in Table (3).

Table 2: The name and the search space of test functions. *"Table 2 continued"*

Test Function	Dimension	Function name
$f_0(x) = \sum_{i=1}^{D} x_i^2$	$[-100,100]^D$	Sphere
$f_1(x) = \sum_{i=1}^{D} \left(x_{i=1}^2 - 10\cos(2\pi x_i) + 10\right)$	$[-5.12,5.12]^D$	Rastrigin
$f_2 \sum_{i=1}^{D} \left(100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2\right)$	$[-50,50]^D$	Rosenbrock
$f_3(x) = \sum_{i=1}^{D} \left(\lfloor x_i + o.5 \rfloor\right)^2$	$[-100,100]^D$	Step
$f_4(x) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D} x_i^2}} - e^{\frac{1}{2}\sum_{i=1}^{D}\cos(2\pi x_i)}$	$[-32,32]^D$	Ackly
$f_5(x) = \left(\sum_{i=1}^{D} \lvert x_i \rvert \exp\left[-\sin\left(x_i^2\right)\right]\right)$	$[-2\pi,2\pi]^D$	Xin-she Yang
$f_6(x) = \sum_{i=1}^{D} \lvert x_i \rvert - \prod_{i=1}^{D} \lvert x_i \rvert$	$[-10,10]^D$	Schwefel's P2.22

Table3: Comparison of the standard Firefly Algorithm and the proposed FAFA on seven functions with 100 iteration numbers. *"Table 3 continued"*

Functions	Dim	Criteria	FA(functions interval)	FA(interval [0,1])	**FAFA(function interval)**	**FAFA(interval [0,1])**
Sphere	10	Best	9.4152e+003	0.0142	**1.6055e-020**	**2.9685e-021**
		Ave-std	1.5147e+004±3.3174e+003	0.0447±0.0218	**4.2587e-007±1.9651e-006**	**2.4765e-007±8.3411e-007**
		Worst	2.3354e+004	0.0970	**1.0789e-005**	**4.4463e-006**
	20	Best	3.0916e+004	0.7595	**1.2260e-020**	**1.2928e-020**
		Ave-std	3.9737e+004±4.8377e+003	1.0226±0.1302	**1.0570e-005±4.8581e-005**	**2.7011e-006±8.4115e-006**
		Worst	4.8619e+004	1.3089	**2.6662e-004**	**4.0106e-005**
	30	Best	4.5983e+004	1.4189	**2.8522e-020**	**1.9438e-018**
		Ave-std	6.7567e+004±7.5095e+003	1.7089±0.1567	**1.4409e-006± 5.5775e-006**	**1.3717e-005± 5.7185e-005**
		worst	7.9282e+004	2.0676	**2.9071e-005**	**3.0144e-004**
Rastrigin	10	Best	67.8467	1.1873	**0**	**0**
		Ave-std	90.5947±8.0385	7.8782±6.0765	**3.2990e-010±1.0685e-009**	**4.7956e-012±1.5098e-011**
		Worst	108.4848	30.6369	**5.5293e-009**	**6.4304e-011**
		Best	178.5922	80.1780	**0**	**0**

	20	Ave-std	210.0403±17.6512	114.6116±14.4153	**4.9036e-008±2.3018e-007**	**9.4311e-009±1.8291e-008**
		Worst	248.9406	143.8536	**1.2642e-006**	**7.8103e-008**
	30	Best	331.5744	148.9788	**0**	**1.9438e-018**
		Ave-std	331.5744±23.3707	197.3660±20.8433	**1.6389e-008±5.3970e-008**	**1.3717e-005± 5.7185e-005**
		Worst	376.1470	223.7772	**2.7912e-007**	**3.0144e-004**
Rosenbrock	10	Best	1.1224e+007	3.2118	**4.5605e-021**	**6.9699e-019**
		Ave- std	2.7801e+008±1.0849e+008	10.3376± 2.9158	**1.5872e-006±4.8986e-006**	**7.3491e-006±3.1143e-005**
		Worst	4.3445e+008	15.7185	**2.2885e-005**	**1.6812e-004**
	20	Best	5.4487e+008	88.1546	**1.7107e-019**	**8.2626e-019**
		Ave-std	1.0279e+009± 2.7068e+008	114.1301±13.4290	**2.9830e-004±0.0014**	**1.4138e-005±3.9789e-005**
		Worst	1.6614e+009	139.8596	**0.0074**	**1.9119e-004**
	30	Best	1.3163e+009	162.4240	**2.8103e-024**	**4.3204e-017**
		Ave-std	2.0430e+009±3.1974e+008	196.0592±15.4955	**2.9366e-004± 0.0011**	**7.1349e-005±2.6516e-004**
		Worst	2.6697e+009	234.4083	**0.0056**	**0.0015**
Step	10	Best	6837	0	**0**	**0**
		Ave-std	1.4779e+004± 3.9020e+003	0±0	**0±0**	**0±0**
		Worst	22165	0	**0**	**0**
	20	Best	27853	0	**0**	**0**
		Ave-std	4.0317e+004±5.4525e+003	0±0	**0±0**	**0±0**
		Worst	48582	0	**0**	**0**
	30	Best	46728	0	**0**	**0**
		Ave-std	6.5004e+004± 7.1381e+003	0±0	**0±0**	**0±0**
		Worst	74335	0	**0**	**0**
Ackly	10	Best	17.8613	0.1623	**5.1115e-012**	**4.7922e-011**
		Ave-std	19.7888± 0.5043	0.3878± 0.1835	**4.3765e-005±1.7566e-004**	**7.9125e-005±1.6136e-004**
		Worst	20.3866	0.8273	**9.5946e-004**	**7.6547e-004**
	20	Best	19.8314	1.8955	**8.5065e-011**	**1.7450e-011**
		Ave-std	20.3897± 0.2070	2.1728± 0.1501	**13.6930±8.5460**	**3.2261e-005±6.2641e-005**
		Worst	20.7067	2.4258	**19.9668**	**2.9809e-004**
	30	Best	19.5064	2.1683	**4.5755e-011**	**5.5674e-011**
		Ave-std	20.4784± 0.2704	2.3626±0.0986	**8.8723e-005±1.9543e-004**	**1.8390e-005±5.4922e-005**
		Worst	20.7544	2.6073	**7.4608e-004**	**2.3851e-004**
Xin- she Yang	10	Best	0.0030	5.8454e-004	**5.6615e-004**	**1.8819e-010**
		Ave-std	0.0092± 0.0042	6.6236e-004± 6.0163e-005	**6.6433e-004±2.6901e-004**	**2.9905e-004±2.5515e-004**
		Worst	0.0230	8.3120e-004	**0.0019**	**5.6607e-004**
	20	Best	8.1151e-006	2.4096e-007	**5.1556e-008**	**5.1533e-008**
		Ave-std	7.0079e-005± 6.9402e-005	7.0934e-007± 3.1021e-007	**8.2643e-008±7.2978e-008**	**3.7632e-007±8.7976e-008**
		Worst	2.8754e-004	1.4285e-006	**3.9944e-007**	**3.9944e-007**
	30	Best	3.7853e-008	1.2688e-011	**3.5141e-012**	**3.5536e-012**
		Ave-std	4.9921e-007± 5.3212e-007	2.1808e-011± 1.2440e-011	**5.7061e-012± 3.1390e-012**	**2.9438e-011± 1.0000e-011**

		Worst	2.2582e-006	6.9601e-011	**1.3425e-011**	**3.3780e-011**
Schwefel 2.22	10	Best	28.4279	0.2143	**8.7864e-012**	**1.6794e-011**
		Ave-std	38.5352±5.7868	0.4634± 0.1498	**1.2029e-006±2.5034e-006**	**1.4800e-006±3.9035e-006**
		Worst	52.1154	0.9226	**9.0389e-006**	**1.9473e-005**
	20	Best	264.8338	2.7295	**5.7054e-011**	**1.5454e-010**
		Ave-std	1.3873e+005± 2.5853e+005	3.6118± 0.3086	**1.2195e-005± 2.1412e-005**	**1.9968e-005± 3.7954e-005**
		Worst	1.0582e+006	4.2449	**7.2401e-005**	**1.8363e-004**
	30	Best	5.6357e+005	5.3047	**2.2082e-010**	**4.0553e-011**
		Ave-std	1.5524e+010± 5.4053e+010	5.9468± 0.3034	**3.7488e-005±7.1676e-005**	**4.7111e-005±1.7124e-004**
		Worst	6.6138e+009	6.7078	**3.0030e-004**	**9.3991e-004**

As illustrated in Table (3), initialization of algorithm is so influential on the final obtained results of algorithm. The final results of Standard FA Algorithm Are far apart in two-mode initialization. As can be observed, the results of the proposed algorithm with two different initial conditions are almost identical to each other. The final results indicate that, tuning the algorithms' parameter by using fuzzy engine was able to fix initializing the algorithm dependent. The proposed algorithm's results are far from the results of standard Firefly Algorithm. The results of the FAFA in all cases are better than FA in 10, 20 and 30 dimensional spaces, and it shows the capability of the proposed algorithm in finding optimal solution. Also, the final results of six functions in 30 dimensional spaces shown in Fig. (4).

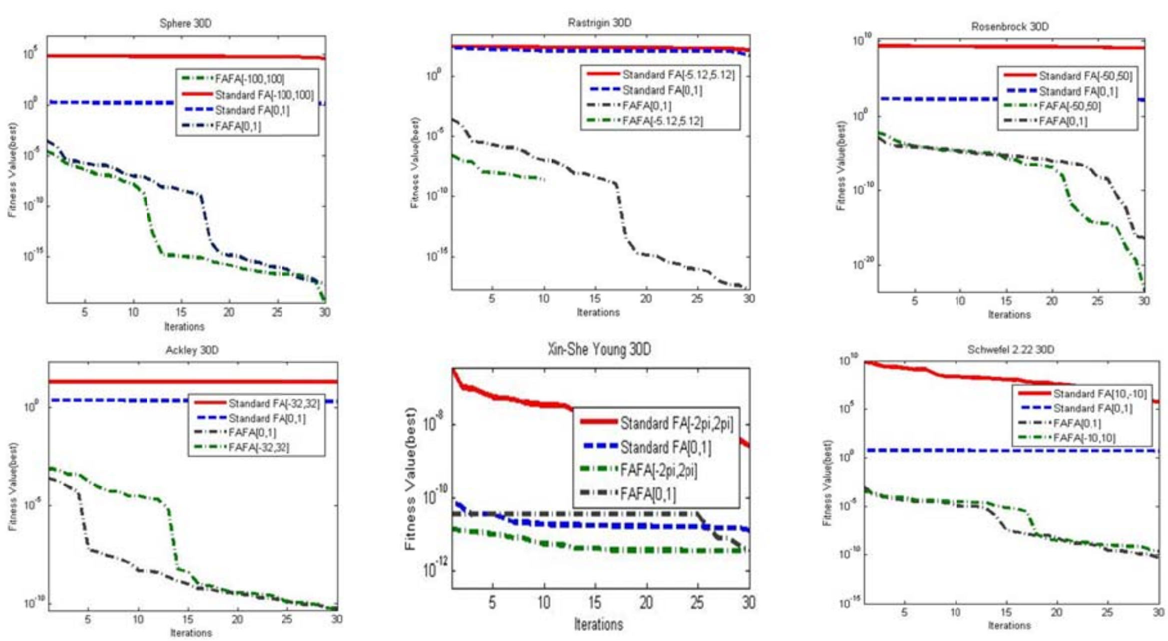


Fig. 4: Comparison of the best results obtained from 30 performances of the standard FA and the

proposed FAFA on different benchmark functions with different initializations on 30-dimensional space for 100 iterations.

The diagrams show that, with increasing the number of iterations results also greatly improved .With increasing the number of iterations, the proposed algorithm succeeded to find the answer of some functions because its parameters tunes successfully. Also, standard Firefly algorithm and proposed algorithm were performed with 500 inner iterations and the results shown in Table (4). As can be seen, besides of function interval, iteration number play an important rule on the accuracy of FA and FAFA as well. Proposed algorithm, enhanced the results significantly.

Table 4: Comparison of the standard Firefly Algorithm and the proposed FAFA on seven functions with 500 iteration number. *"Table 4 continued"*

Functions	Dim	Criteria	FA(functions interval)	FA(interval [0,1])	FAFA(function interval)	FAFA(interval [0,1])
Sphere	10	Best	7.8232e+003	0.0043	**4.2468e-081**	**1.7855e-081**
		Ave-std	1.7465e+004±3.3935e+003	0.0240± 0.0170	**3.8031e-006±2.0831e-005**	**6.6332e-020± 3.6331e-019**
		Worst	2.2767e+004	0.0916	**1.1409e-004**	**1.9900e-018**
	20	Best	2.9065e+004	0.7888	**1.3459e-081**	**2.0415e-082**
		Ave-std	3.9237e+004± 5.0170e+003	1.0151±0.1169	**1.1109e-063±3.5024e-063**	**1.8623e-008±1.0200e-007**
		Worst	4.7632e+004	1.2996	**1.7832e-062**	**5.5870e-007**
	30	Best	4.9231e+004	1.1747	**2.9320e-080**	**1.0688e-081**
		Ave-std	6.8124e+004±7.9629e+003	1.6241±0.1908	**3.3763e-009±1.8493e-008**	**2.5890e-017±1.4181e-016**
		worst	8.3782e+004	1.9414	**1.0129e-007**	**7.7671e-016**
Rastrigin	10	Best	0.0117	0.2084	**0**	**0**
		Ave-std	15.3828±11.0676	1.5197±0.8532	**0±0**	**0±0**
		Worst	40.7697	3.1869	**0**	**0**
	20	Best	0.3430	8.9889e-005	**0**	**0**
		Ave-std	24.4577±16.6836	3.2827±6.8467	**0±0**	**3.3928e-012±1.8583e-011**
		Worst	79.4104	19.9477	**0**	**1.0179e-010**
	30	Best	7.4961e-006	1.0386e-007	**0**	**0**
		Ave-std	0.6322±2.9602	0.0035±0.0059	**0±0**	**0± 0**
		Worst	16.1476	0.0264	**0**	**0**
Rosenbrock	10	Best	6.4759e+007	2.4930	**0**	**0**
		Ave- std	2.4602e+008±1.2064e+008	9.0253±3.0483	**7.8615e-021±4.3052e-020**	**4.5739e-013±2.5049e-012**
		Worst	4.8834e+008	15.9178	**2.3581e-019**	**1.3720e-011**
	20	Best	3.4543e+008	81.4058	**0**	**0**
		Ave-std	1.0000e+009±2.2424e+008	116.8403±12.2185	**0.0047± 0.0246**	**0±0**
		Worst	1.3529e+009	138.0214	**0.1347**	**0**
	30	Best	9.4882e+008	164.0642	**0**	**0**
		Ave-std	1.8556e+009±3.4365e+008	189.3134±11.8458	**4.3658e-005± 2.3912e-004**	**1.2925e-026±7.0792e-026**
		Worst	2.5855e+009	212.3674	**0.0013**	**3.8774e-025**
Step	10	Best	11270	0	**0**	**0**
		Ave-std	1.5534e+004±2.6704e+003	0±0	**0±0**	**0±0**
		Worst	22372	0	**0**	**0**
	20	Best	23664	0	**0**	**0**
		Ave-std	3.9463e+004±5.1446e+003	0±0	**0±0**	**0±0**
		Worst	49443	0	**0**	**0**
	30	Best	48362	0	**0**	**0**
		Ave-std	6.6659e+004±7.2928e+003	0±0	**0±0**	**0±0**
		Worst	83095	0	**0**	**0**
Ackly	10	Best	18.3739	0.1330	**-8.8818e-016**	**-8.8818e-016**
		Ave-std	19.6831±0.3862	0.3230±0.1361	**3.1707e-004±0.0017**	**9.2557e-007±5.0068e-006**
		Worst	20.3177	0.7174	**0.0095**	**2.7433e-005**

	20	Best	19.8637	1.9657	**1.5877e-004**	**-8.8818e-016**
		Ave-std	20.2739±0.2164	2.1999±0.1174	**16.7160±6.1649**	**1.2552e-009±5.5966e-009**
		Worst	20.7337	2.4971	**19.9668**	**2.9933e-008**
	30	Best	20.0047	2.1956	**0.1080**	**1.0214e-016**
		Ave-std	20.5218±0.1616	2.3347±0.0906	**6.4292±5.5210**	**1.2143e-016±3.6181e-016**
		Worst	20.7452	2.5102	**17.8851**	**2.6645e-015**
Xin- she Yang	10	Best	8.7880e-004	5.6611e-004	**5.6607e-004**	**2.4161e-041**
		Ave-std	0.0019±5.0232e-004	5.7618e-004±7.9985e-006	**7.3795e-004±5.5958e-004**	**3.7081e-004±2.3182e-004**
		Worst	0.0029	5.9507e-004	**0.0035**	**5.6607e-004**
	20	Best	2.1355e-007	5.1536e-008	**5.1559e-008**	**5.1533e-008**
		Ave-std	4.3521e-007±2.5167e-007	9.0842e-008±3.6588e-008	**8.5274e-008±7.2414e-008**	**3.0683e-007±1.5621e-007**
		Worst	1.1029e-006	1.6001e-007	**3.9944e-007**	**3.9944e-007**
	30	Best	3.5137e-012	3.5124e-012	**3.5131e-012**	**3.5177e-012**
		Ave-std	8.2427e-012± 2.7922e-012	3.5125e-012±7.6790e-017	**7.5190e-012±6.6262e-012**	**2.6843e-011±1.2795e-011**
		Worst	1.2908e-011	3.5128e-012	**3.3780e-011**	**3.3780e-011**
Schwefel 2.22	10	Best	20.3051	0.2162	**1.0475e-041**	**2.4583e-041**
		Ave-std	30.1988±4.1126	0.3889±0.1103	**3.7519e-010±2.0550e-009**	**1.4191e-006±6.6634e-006**
		Worst	36.8975	0.6708	**1.1256e-008**	**3.6525e-005**
	20	Best	57.3202	0.3268	**3.1408e-041**	**1.9184e-041**
		Ave-std	693.9269±1.8023e+003	3.4604±0.6785	**1.7001e-010±9.1855e-010**	**1.1485e-012±4.9095e-012**
		Worst	9.7824e+003	4.1321	**5.0331e-009**	**2.6037e-011**
	30	Best	46.4510	0.0024	**8.9663e-042**	**3.1949e-042**
		Ave-std	1.0231e+003±4.9131e+003	2.6031±2.3962	**4.7391e-019± 1.8647e-018**	**3.2669e-006±1.7894e-005**
		Worst	2.7035e+004	6.0330	**9.9950e-018**	**9.8008e-005**

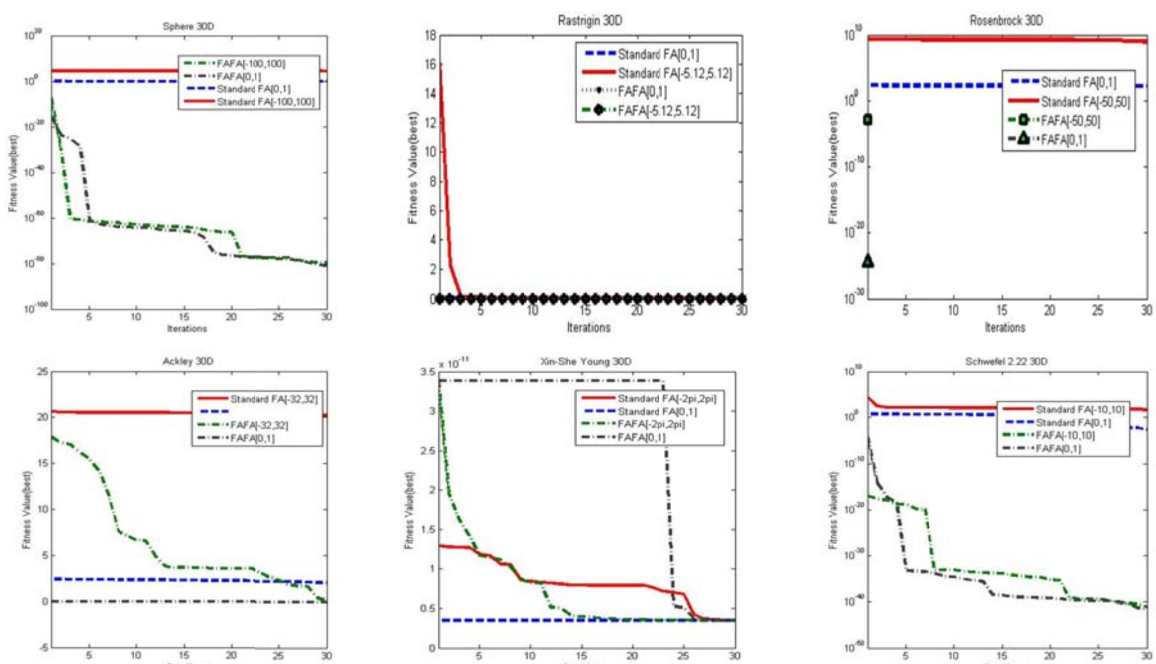


Fig. 5: Comparison of the best results obtained from 30 performances of the standard FA

and the proposed FAFA on different benchmark functions with different initializations on 30-dimensional space for 500 iterations.

In the Fig. (5), the best results of six functions in 30 dimensional spaces are shown. As it shown, in the early repetitions, the error rate of algorithm is notable but during the next iterations it's improved considerably. Results proved that, algorithm parameters are so effective on the algorithm performance.

In addition, the results of proposed algorithm compared with Fuzzy FA[19], which is another version of using fuzzy logic to improve original FA and CLA-FA[20]. All of these algorithms performed in 10, 20 and 30 dimensional spaces 30 times with same initial parameters that mentioned in the beginning of this section. The results of FAFA, Fuzzy FA and CLA-FA illustrated in Table 5 after 500 internal iterations. As it shown new proposed algorithm is outperform of Fuzzy FA and CLA-FA in the most cases and it increased the performance of FA steadily. In the fuzzy FA the search strategy of FA and in FAFA its parameters analyzed. The results proved that the role of parameters in FA is more noticeable in compare of its search strategy.

Table 5: Comparison of the FAFA, Fuzzy FA and CLA-FA on seven functions with 500 iteration number. *"Table 5 continued"*

Functions	Dim	Criteria	CLA-FA(function interval)	Fuzzy FA(function interval)	**FAFA(function interval)**
Sphere	10	Best	8.0183e-010	1.5713e-009	**4.2468e-081**
		Ave-std	1.1183e-005±1.0072e-004	0.0044e-002 ±0.0020e-003	**3.8031e-006±2.0831e-005**
		Worst	7.0004e-004	0.0002	**1.1409e-004**
	20	Best	3.7311e-009	4.0267e-0013	**1.3459e-081**
		Ave-std	1.9209e-004±2.3369e-004	1.9002e-005 ±2.0068e-006	**1.1109e-063±3.5024e-063**
		Worst	0.0003	0.0010e-004	**1.7832e-062**
	30	Best	1.8423e-008	5.9003e-008	**2.9320e-080**
		Ave-std	5.2341e-005±2.0079e-004	0.0017-e004 ±0.0004	**3.3763e-009±1.8493e-008**
		worst	0.0013e-001	1.0295	**1.0129e-007**
Rastrigin	10	Best	0.0020e-003	4.1004e-0010	**0**
		Ave-std	0.0400e-001±0.1381e-001	1.4200e-008 ±1.2370e-009	**0±0**
		Worst	0.0084	7.9224e-0012	**0**
	20	Best	0.2114e-005	4.7334e-0012	**0**
		Ave-std	0.1988e-004±1.0095e-003	1.7817e-008 ±2.0063e-009	**0±0**
		Worst	2.4038e-001	0.0010e-001	**0**
	30	Best	0.0216e-10	1.2830e-10	**0**
		Ave-std	1.1001e-004±0.0601e-4	1.1557e-014 ±2.1096e-014	**0±0**
		Worst	0.0368e-001	3.0623e-006	**0**
Rosenbrock	10	Best	8.8071e-009	4.8700e-009	**0**
		Ave- std	0.0347e-003 ±1.0074e-002	0.0005e-004 ±0.0024e-05	**7.8615e-021±4.3052e-020**
		Worst	3.0177e-001	0.0130e-001	**2.3581e-019**
	20	Best	6.0295e-013	3.0295e-020	**0**
		Ave-std	0.0350e-005 ±0.0022e-005	0.0050e-004 ±0.0022e-005	**0.0047± 0.0246**
		Worst	0.6543e-004	0.0050e-002	**0.1347**
	30	Best	3.5007e-007	4.5068e-013	**0**
		Ave-std	1.9817e-003 ±1.9032e-002	0.0017e-005 ±0.0012e-004	**4.3658e-005± 2.3912e-004**
		Worst	1.1245e-001	0.0033	**0.0013**
Step	10	Best	0	0	**0**
		Ave-std	0±0	0±0	**0±0**
		Worst	0	0	**0**
	20	Best	0	0	**0**
		Ave-std	0±0	0±0	**0±0**
		Worst	0	0	**0**

	30	Best	0	0	**0**
		Ave-std	0±0	0±0	**0±0**
		Worst	0	0	**0**
Ackly	10	Best	0.7643e-006	9.0098e-008	**-8.8818e-016**
		Ave-std	0.0187e-003±0.0138e-003	0.0019 ±0.0014	**3.1707e-004±0.0017**
		Worst	0.7622e-002	0.0100	**0.0095**
	20	Best	0.0060e-006	2.8020e-009	**1.5877e-004**
		Ave-std	0.8169e-003±1.0064e-002	0.0012e-002 ±0.0011e-003	**16.7160±6.1649**
		Worst	0.0387e-001	0.9041e-003	**19.9668**
	30	Best	1.0076e-0010	3.5211e-0012	**0.1080**
		Ave-std	2.0121e-005±0.7037e-004	0.0011e-005 ±0.0010e-004	**6.4292±5.5210**
		Worst	2.0240e-002	0.1034	**17.8851**
Xin- she Yang	10	Best	2.6752e-006	2.2364e-004	**5.6607e-004**
		Ave-std	6.9936e-005±1.0081e-005	5.4339e-014 ±8.6301e-015	**7.3795e-004±5.5958e-004**
		Worst	5.1160e-004	5.1108e-014	**0.0035**
	20	Best	2.1008e-008	2.1533e-012	**5.1559e-008**
		Ave-std	6.8820e-008±4.1323e-009	7.9927e-009 ±8.0067e-009	**8.5274e-008±7.2414e-008**
		Worst	2.1066e-008	3.9004e-007	**3.9944e-007**
	30	Best	2.8317e-012	3.5124e-012	**3.5131e-012**
		Ave-std	4.5380e-012±4.8745e-013	4.5214e-012 ±5.5261e-012	**7.5190e-012±6.6262e-012**
		Worst	2.1186e-011	3.3780e-015	**3.3780e-011**
Schwefel 2.22	10	Best	1.9870e-004	2.6460e-009	**1.0475e-041**
		Ave-std	1.1092e-003 ±1.1032e-002	0.0022 ±0.0022e-001	**3.7519e-010±2.0550e-009**
		Worst	1.2079e-002	0.0079e-002	**1.1256e-008**
	20	Best	0.3751e-005	1.4753e-005	**3.1408e-041**
		Ave-std	0.2886e-004 ±0.0910e-004	0.0333e-003 ±0.0020e-004	**1.7001e-010±9.1855e-010**
		Worst	0.0271e-002	0.0050e-004	**5.0331e-009**
	30	Best	1.5672e-008	3.5672e-008	**8.9663e-042**
		Ave-std	0.0020e-007 ±1.9821e-005	0.1120 ±0.0021e-001	**4.7391e-019± 1.8647e-018**
		Worst	2.1591e-002	0.0032	**9.9950e-018**

5. Otsu Method

The basic idea of the Otsu method is to divide pixels into two groups according to a threshold value. All the pixels that are lower than the specified threshold will be in first group and pixels that are greater than the threshold will be put in the another group. The Otsu method uses the variance within class and variance between classes to determine thresholds. Otsu introduced a method which maximizes the between class variance and minimizes the within class variance to separate the segmented classes as farther as possible in order to find the optimal threshold for segmentation. This between class variance measures is called Otsu's multi-threshold measure and is used as the objective function in the proposed FAFA. It is formulated as follows.

The image size is $M \times N$ and the image gray level is L. The gray range is $0\sim L-1$. The pixels' number of grayscale level i is n_i. Thus, the number of the image pixels is $n = \sum_{i=0}^{l-1} n_i = M \times N$, normalized Histogram, the probability distribution is:

$$p_i = \frac{n_i}{n}, \sum_{i=0}^{L-1} p_i = 1 \tag{7}$$

The image is divided into two classes with the standard threshold t. The class c_1 includes the pixels $i \leq t$ and the class c_2 involves the pixels $i \geq t$.

Cumulative probability of c_1 and c_2 is:

$$\omega_1 = \sum_{i=0}^{t} p_i = \omega(t) \tag{8}$$

$$\omega_2 = \sum_{i=t+1}^{L-1} p_i = 1 - \omega(t) \tag{9}$$

The mean levels are calculated as:

$$\mu_1 = \sum_{i=0}^{t} \frac{ip_i}{\omega_1} = \sum_{i=0}^{t} \frac{ip_i}{\omega(t)} \tag{10}$$

$$\mu_2 = \sum_{i=t+1}^{L-1} \frac{ip_i}{\omega_2} = \sum_{i=t+1}^{L-1} \frac{ip_i}{1-\omega(t)} \tag{11}$$

Variances of class c_1 and class c_2 are:

$$\sigma_1^2 = \sum_{i=0}^{t} \frac{(i-\mu_1)^2 p_i}{\omega_1} \tag{12}$$

$$\sigma_2^2 = \sum_{i=t+1}^{L-1} \frac{(i-\mu_2)^2 p_i}{\omega_2} \tag{13}$$

Inter-cluster variance is:

$$\sigma_\omega^2 = \omega_1\sigma_1^2 + \omega_2\sigma_2^2 \tag{14}$$

Intra-cluster variance is:

$$\sigma_B^2 = \omega_1(\mu_1-\mu_T)^2 + \omega_2(\mu_2-\mu_T)^2 = \omega_1\omega_2(\mu_2-\mu_1)^2 \tag{15}$$

It must be noted that the μ_T is the mean of the whole image.

The best threshold value T should satisfy the Eq. (16) after the image is divided into two categories c_1 and c_2:

$$\eta\,|_T = \max\left[\frac{\sigma_B^2}{\sigma_w^2}\right] \tag{16}$$

When it segments the image with complex target area, the Otsu method should be extended to two or more thresholds. In this case, according to the Eq. (17) and (18), inter cluster variance and intra-cluster variance of each cluster must be calculated. C is the number of clusters.

$$\sigma_w^2 = \omega_1\sigma_1^2 + \omega_2\sigma_2^2 + \ldots + \omega_c\sigma_c^2$$

(17)

$$\sigma_B^2 = \omega_1(\mu_1-\mu_T)^2 + \omega_1(\mu_2-\mu_T)^2 + \ldots + \omega_3(\mu_3-\mu_T)^2$$

(18)

$$\eta\Big|thresholds = \max\left[\frac{\sigma_B}{\sigma_\omega}\right] \tag{19}$$

Eq. (19) is used as objective function for the proposed FAFA based procedure which is to be maximized.

6. Proposed Segmentation Method

In this section, a hybrid segmentation method based on Otsu method and the proposed Fuzzy Adaptive Firefly Algorithm is developed. The fireflies Vector-dimension number depends on the number of thresholds we use in segmentation. When it is initialized, we spread fireflies over the whole plan randomly. At the same time, we obtain the fitness function of each firefly subsequently, update the fireflies' fitness and position according to the proposed Firefly Algorithm.

The details of the proposed segmentation algorithm are introduced as follows:

Step 1: Initialize fireflies with random threshold values and set FAFA parameters.

Step 2: Segment the image according to threshold values.

Step 3: Calculate the fitness function of image according to Eq. (19).

Step 4: Move firefly to the new position in a way that it maximizes Eq. (19) and updates the fireflies' position and fitness.

Step 5: Judge whether the iteration number has reached the maximum value. If so, deduction ends, otherwise, jump to the Step 2.

Step 6: Finally, images are segmented by optimal thresholds.

The six steps of the proposed segmentation method can be illustrated as the following flowchart in Fig. (6).

Fig. 6: The flowchart of the proposed segmentation method.

We use uniformity measure to evaluate segmentation results and compare them with the results of different methods.

$$U = 1 - \left(2 \times c \times K / N / \left(f_{\max} - f_{\min}\right)^2\right) \tag{20}$$

where,
c is the number of thresholds.
K is the sum of all clusters' variances as illustrated in Eq. (21).
N is the total number of pixels in the given image.
f_{max} is the maximum gray level of pixels in the given image.
f_{min} is the minimum gray level of pixels in the given image.

$$K = \sum_{i=0}^{c} \sum_{i \in R_i} \left(f_i - \mu_j\right)^2 \tag{21}$$

where,

R_j is the jth segmented region.
f_i is a gray level of the pixel i.
μ_j is the mean gray level of the pixels in jth region.
At the end, the uniformity measure can be presented as Eq. (22).

$$U = 1 - 2 \times c \times \sum_{i=0}^{c} \sum_{i \in R_i} (f_i - \mu_j)^2 / N / (f_{\max} - f_{\min})^2 \qquad (22)$$

The value of the uniformity measure, U, should be a positive fraction, i.e. it should lie between 0 and 1. A higher value of U indicates that there is better uniformity in the thresholded image, depicting better quality of thresholding. Conversely, a lower value of U indicates worse quality of the thresholding procedure.

7. Experimental Results of the Proposed Segmentation Method

In this experiment, the Population size is 25. The fireflies are initialized as a gray value within the interval of (0 ~ 255) and the largest truncated generation is 100. The dimension of input vectors depends on the number of thresholds we use for segmentation. In this paper, we use different threshold numbers, from 2 to 5, for segmentation.

We test the proposed algorithm on the benchmark images of ‘Lena’, ‘Peppers’, ‘Bird’, ‘Cameraman’, ‘House1’, ‘House2’, ‘Elaine’ and ‘Gold hill’. These original test images are illustrated in Fig. (7).

Table (6) shows the uniformity values of FAFA-Otsu, FA-Otsu[21], PSO-Otsu and SMCAPSO[22] on eight test images with different thresholds. As demonstrated, the results of proposed method are better than the three other methods in all cases. It shows that new improved FA can be effective on image segmentation as well.

Lena (512*512) Peppers (512*512) Bird (512*512) Cameraman (512*512)

Goldhill(512*512) House1(512*512) House2(512*512) Elaine(512*512)

Fig. 7:The test images: ‘Lena’, ‘Peppers’, ‘Bird’, ‘Cameraman’, ‘Gold hill ’, ‘House1’,

'House2' and 'Elaine'.

Table (6) shows the uniformity values of FAFA-Otsu, FA-Otsu[20], PSO-Otsu and SMCAPSO[21] on eight test images with different thresholds. As demonstrated, the results of proposed method are better than the three other methods in all cases. It shows that new improved FA can be effective on image segmentation as well.

Table 6: Comparison of uniformity for FAFA-Otsu, FA-Otsu, PSO-Otsu and SMCAPSO methods.

Image	c	Uniformity			
		FAFA-Otsu	FA-Otsu	PSO-Otsu	SMCAPSO
Lena (512*512)	2	0.9612	0.9472	0.8858	0.9694
	3	0.9775	0.9412	0.8861	0.9587
	4	0.9790	0.9645	0.8899	0.9541
	5	0.9833	0.9790	0.8993	0.9843
Peppers (512*512)	2	0.9604	0.9454	0.6703	0.9565
	3	0.9737	0.9512	0.6786	0.9768
	4	0.9741	0.9612	0.6952	0.9739
	5	0.9752	0.9778	0.7054	0.9733
Bird (512*512)	2	0.9783	0.9612	0.7551	0.9750
	3	0.9799	0.9639	0.7625	0.9760
	4	0.9819	0.9659	0.7689	0.9800
	5	0.9830	0.9701	0.7691	0.9805
Camera (512*512)	2	0.9824	0.9724	0.7564	0.9727
	3	0.9841	0.9715	0.7588	0.9773
	4	0.9879	0.9732	0.8107	0.9777
	5	0.9883	0.9737	0.8182	0.9779
House1(512*512)	2	09950	0.9940	0.8472	0.8427
	3	0.9954	0.9945	0.8333	0.8489
	4	0.9956	0.9950	0.8263	0.8596
	5	0.9965	0.9949	0.8090	0.9043
House2(512*512)	2	0.9953	0.9947	0.8517	0.8652
	3	0.9958	0.9961	0.8303	0.8866
	4	0.9951	0.9934	0.7443	0.8391
	5	0.9959	0.9956	0.7865	0.7194
Elaine(512*512)	2	0.9956	0.9933	0.8223	0.8437
	3	0.9966	0.9942	0.7256	0.7771
	4	0.9957	0.9952	0.7789	0.8344
	5	0.9963	0.9962	0.8193	0.8566
Gold hill (512*512)	2	0.9833	0.9623	0.7870	0.9749
	3	0.9854	0.9644	0.7839	0.9764

	4	0.9861	0.9652	0.8049	0.9800
	5	0.9869	0.9671	0.8161	0.9823

Also, in Table 7, we compare the obtained thresholds of the proposed algorithm on eight images in 2 to 5 thresholds level with three other methods.

Table 7: Comparison of the obtained optimal thresholds for FAFA-Otsu, FA-Otsu, PSO-Otsu and SMCAPSO methods.

Methods / Images	C	FAFA-Otsu	FA-Otsu	PSO-Otsu	SMCAPSO
Lena (512*512)	2	74,129	57,118	76,152	63,125
	3	45,105,145	67,82,131	62,102,155	54,100,164
	4	53,98,109,138	53,108,141,158	46,90,120,162	44,74,128,187
	5	42,75,113,147,179	38,72,102,124,167	47,76,105,131,172	35, 83,128,132,184
Peppers (512*512)	2	68,126	89,132	77,127	64,125
	3	68,129,165	58,95,162	65,113,163	78,124,173
	4	49,88,133,175	42,72,106,143	36, 85,126,166	39, 87,135,174
	5	49, 96, 132,161,165	54,82,107,127,169	40,82,116,153,180	36,80,117,128, 184
Bird (512*512)	2	91,121	52,116	74,126	35,117
	3	70,129,168	76,144,167	64,116,159	36,105,123
	4	65,111,150,164	79,115,137,170	65,89,131,162	53,78,113,155
	5	58,73, 112,144,167	43,83,114,144,180	1 4,74,109,144,183	75, 90,104,122,154
Camera (512*512)	2	66,149	74, 127	69,143	57,127
	3	66,122,157	37,71,130	60,128,158	88,124,155
	4	61,107, 134, 160	25, 67,121,150	43,93,137,167	30,74,137,160
	5	37, 80,122,156,175	46,74,129,135,155	33,80,128,152,174	54, 81,130,143,195
House1 (512*512)	2	104,186	108,164	116,194	131,198
	3	100,162,207	75,125,184	96,158,210	111,162,206
	4	95,144,166,215	121,173,231,252	88,127,166,218	108,152,183,234
	5	85,114,149,186,232	24,86,122,150,204	84,117,145,179,223	40,85,136,170,233
House2 (512*512)	2	113,167	120,169	90,172	115,152
	3	104,137,183	96,150,191	75,131,183	92,170,194
	4	67,112,150,179	102,126,170,203	77,132,163,192	28,91,139,191
	5	49,72,125,178,208	61,102,147,189,201	68,106,140,170,196	57,106,148,180,232
Elaine (512*512)	2	119,164	115,146	106,169	122,183
	3	117,142,210	83,109,173	105,141,180	92,156,197
	4	86,105,140,179	16,91,127,169	87,126,158,198	88,111,155,199
	5	84,96,116,163,204	94,108,129,137,168	78,110,137,171,210	80,114,143,164,221
Gold hill (512*512)	2	91,150	78,155	98,157	98,161
	3	72,112,164	58,109,177	89,135,187	86,119,195

	4	77,107,125,174	64,92,142,189	64,104,142,188	95,102,145,200
	5	48,51,89,117,178	58,87,121,128,202	61,88,119,152,201	69, 96,124,131,176

As a subjective evaluation we demonstrate the obtained thresholded images for 'Lena', 'peppers' and 'House1' images with different FAFA-Otsu, FA-Otsu, PSO-Otsu and SMCAPSO methods for 4 and 5 thresholds in Fig. (8). As illustrated, different parts of the image have been set apart according to the obtained thresholds. In the thresholded images, different clusters are depicted by different colors. The pixels are put into different clusters according to their gray level value and the specified threshold values. One indicator of a good segmentation algorithm is the matter that as the number of thresholds increases, the segmented image must be much similar to the original one. As illustrated, this happens in our resulted images.

Thresholds | Method: FAFA-OTSU | FA-OTSU | PSO-OTSU | SMCAPSO

C=4

C=5

C=4

C=5

C=4

C=5

Fig. 8: The obtained segmented images based on the proposed FAFA-Otsu, SMCAPSO methods on 'Lena', 'Peppers' and 'House1' imagees with different threshold levels.

8. Conclusion

In this paper, to improve the local and global searches and to hold a balance between them in Firefly Algorithm, we first presented a Fuzzy Adaptive version of the Firefly Algorithm. We used an adaptive weight to adapt α and γ parameters of FA. Our experimental results on seven standard benchmark functions including Sphere, Rastrigin, Rosenbrock, Step, Ackly Xin-She Yang and Schwefel's P2.22 functions in10, 20 and 30 dimensional spaces indicate that our proposed algorithm improves the performance of FA especially in high dimensional spaces. Then to show this new algorithm ability in other applications we used it for image segmentation and we proposed a hybrid method by using proposed FAFA and Otsu's method to segment the images of 'Lena', 'Peppers', 'Bird', 'Cameraman', 'House1','House2','Elaine' and 'Gold Hill'. The different objective and subjective evaluations done and the results of the proposed segmentation method were compared with different well-known segmentation methods. The results demonstrate a significant enhancement in the performance of the presented method. Finally, it can be concluded that new FAFA in more effective that FA and some other modified version of FA and can be used for different applications.

References

1. T. Baeck, D. B. Fogel and Z. Michalewicz, Handbook of evolutionary computation. Taylor & Francis (1997).
2. E. Bonabeau, M. Dorigo and G. Theraulaz, Swarm intelligence: From natural to artificial systems. Oxford University Press (1999).
3. J. Kennedy, R. Eberhart and Y. Shi, Swarm intelligence. Academic Press (2001).

4. X. S. Yang, Biology-derived algorithms in engineering optimization. (chapter 32)-In: Olarius&Zomaya (eds) Handbook ofbio inspiredalgorithms and applications. Chapman & Hall/CRC (2005).
5. Y. Liu and K. M. Passino, Swarm intelligence: A survey. Proceedings of Fourth International Conference of Swarm Intelligence (2005).
6. J. Kennedy and R. C. Eberhart, Particle swarm optimization. IEEE International Conference on Neural Network (1995) 1942-1948.
7. M. Dorigo, M. Birattari and T. Stutzle, Ant colony optimization. IEEE Computational Intelligent Magazine 1 (2006)28-39.
8. L. X. Li, Z. J. Shao and J. X. Qian, An optimizing method based on autonomous animate: Fish swarm algorithm. Proceedings of System Engineering Theory and Practice. 11(2002)32-38.
9. D. Karaboga and B. Basturk, On the performance of artificial bee colony algorithm. Applied Soft Computing. (2008) 687-697.
10. X. S. Yang, Nature-inspired etaheuristic algorithms. Luniver Press (2008).
11. X. S. Yang, Firefly algorithm, levy flight and global optimization. In: Brame M, Elli R, Petridis M (eds) Research and development in intelligent system XXVI.Springer, London, (2010) 209-218.
12. M. M. Sayadi, R. Ramezanian and N. Ghafari-Nasb, A discrete firefly metaheuristic with local search for makespan minimization in permutation flow shop scheduling problem. International Journal of Industrial Engineering Computations. 1 (2010) 1-10.
13. M. H. Horng and T. W. Jiang, The codebook design of image vector quantization based on the firefly algorithm. Computational Collective Intelligence, Technologies and Applications, LNCS 6423 (2010) 438-447.
14. M. H. Horng, Vector quantization using the firefly algorithm for image compression. Expert Systems with Applications 38, (article in press) 12 Aug (2011).
15. M. H. Horng and R. J. Liou, Multilevel minimum cross entropy threshold selection based on the firefly algorithm. Expert Systems with Applications. 38 (2011) 14805-14811.
16. M. H. M. Noor, A. R. Ahmad, Z. Hussain, K. A. Ahmad and A. R. Ainihayati, Multilevel thresholding of gel electrophoresis images using firefly algorithm. Proceedings of Control System, Computing and Engineering (2011) 18-21.
17. X. S. Yang, Firefly algorithm for multimodal optimization Stochastic Algorithms: Foundations and Applications, SAGA, lecture notes in computer sciences. 5792 (2009) 169-178.
18. N. Otsu N, A threshold selection method from gray-level histogram..IEEE Trans Systems Man Cybern 9(1) (1979) 62–66.
19. T. Hassanzadeh, H. Vojodi and A. M. E. Moghadam, An image segmentation approach based on maximum variance intra-cluster method and firefly algorithm. Proceeding of Seventh International Conference on Natural Computation. (2011) 1844-1848.
20. T. Hassanzadeh, and M. R. Meybodi, A New Hybrid Algorithm Based on Firefly Algorithm and Cellular Learning Automata, 20th Iranian Conference on Electrical Engineering (2012).

21. A. Chander, A. Chatterjee and P. Siarry, A new social and momentum component adaptive PSO algorithm for image segmentation. Expert Systems with Applications. 38 (2011) 4998-5004.
22. T. Hassanzadeh, H. R. Rashidi Kanan, Fuzzy FA: a modified firefly algorithm. Applied Artificial Intelligence. 28 (2014) 47-65.