A Multimodal Firefly Optimization Algorithm Based on Coulomb’s Law

TAYMAZ RAHKAR-FARSHI
Computer engineering
Istanbul Aydin University
Sefaköy-Küçükçekmece, Istanbul
TURKEY

Abstract: - In this paper, a multimodal firefly algorithm named the CFA (Coulomb Firefly Algorithm) has been presented based on the Coulomb’s law. The algorithm is able to find more than one optimum solution in the problem search space without requiring any additional parameter. In this proposed method, less bright fireflies would be attracted to fireflies which are not only brighter, but according to the Coulomb’s law pose the highest gravity. Approaching the end of iteration, fireflies' motion steps are reduced which finally results in a more accurate result. With limited number of iterations, groups of fireflies gather around global and local optimal points. After the final iteration, the firefly which has the highest fitness value, would be survived and the rest would be omitted. Experiments and comparisons on the CFA algorithm show that the proposed method has successfully reacted in solving multimodal optimization problems.

Key-Words: - Swarm Intelligence, multimodal firefly algorithm, multimodal optimization, firefly algorithm

1 Introduction

Optimization is finding an optimum solution from a set of available options with the purpose of optimizing criteria for the problem in a limited time. The main challenge with single solution optimization algorithms, however, is that they are only able to find one optimum solution from a set of available options while most real-world problems have more than one optimum solution [1]. Hence, multimodal optimization algorithms which are among the novel inventions of evolutionary algorithms, have been designed to find a set of possible solutions from available options. Unlike unimodal optimization algorithms which try to avoid local optimal points, multimodal optimization algorithms recognize these points as a solution. Although normally the algorithms have not been basically designed to merely solve these problems, several algorithms have recently tried to solve these problems by modifying existing unimodal optimization algorithms. The majority of these algorithms are based on particle swarm optimization algorithms [1-6] and genetic algorithms [7-10]. The firefly optimization algorithm has been used successfully to optimize different kinds of problems, but all of them have been within the span of unimodal optimization problems. In this paper, the Coulomb’s law has been applied to the firefly optimization algorithm in order to turn it into a multimodal algorithm.

2 Related Work

2.1. EPSO

EPSO algorithm [3] was introduced by J. Barbara and Carlos A. C. in 2009. In this method, the selection of global optimum mechanism, in PSO algorithm, was changed using Coulomb's law. Then, the particles that are to be selected as the global optimum can be separately calculated for each particle. In fact, particles may move towards different particles as the global optima. In other words, the global optima for every particle could vary from one particle to another. Hence, particles not only do not surround the global optima, but they also surround their local optima. It is evident that a particle with a more desirable cost function is surrounded by more particles. It is this mechanism's property that particles tend to move towards a point that has both an appropriate cost function value and an appropriate distance from the particle.

2.2. FERPSO

FERPSO [2] is a well-known algorithm that has been proposed for solving multimodal optimization problems which was introduced by Xiaodong Li in 2007. In terms of nature, this algorithm could be viewed as: more birds will gather where there is more food. In fact, if they find a good resource near themselves, they will not use farther resources. In FERPSO, the particles that are to be selected as global optimal point are selected for each particle regarding the Euclidian distance between particles. In essence, the overall structure of FERPSO and EPSO are highly similar, and they both have the same level of complexity.

2.3. LPSO

B. Y. Qu et al have combined a local searching technique with some existing multimodal PSO optimization algorithms that have used niching [2, 11, 12] method trying to solve such problems. In this method, the personal best for particles are improved significantly by using a local searching method. In fact, the personal best is improved by generating a random point between the particle and the nearest point, that is, if the newer point is more desirable than the current personal best, the new point will replace the former, otherwise, the original point stays intact.

2.4. SNPSO

J.Zhang et al. [13] proposed a modified algorithm called the sequential niching particle swarm optimization (SNPSO). This algorithm divides the whole population into several sup-populations which can be located around optimal solutions in multimodal problems. They use space convergence rate (SCR), in which each sub-population detects global and local optimal solutions until the end of iteration.

2.5. SPSO

Xiaodong li. [11] proposed an improved PSO algorithm called the (SPSO). In this method, the idea of species is used to specify each species' best value of neighborhood. The algorithm divides the whole population into several populations called species with regard to their similarity. Each species gather around a particle called species seed.

3 Firefly Algorithm

3.1. The Behavior of Fireflies in Nature

There are almost two thousand known species of firefly in nature, most of which emit flashes of light with a certain rhythm in order to attract a mating partner or bait. In addition to these reasons, fireflies can protect themselves against the attackers using the flashes which can also attract the opposite sexes. The distance between the fireflies and the environment, where the light is emitted, is somehow effective on the intensity of light received by fireflies. As the light intensity obeys the inverse square law at a particular distance r (I ∝ 1/r2), and because light is absorbed by air, most fireflies can just be visible to a limited distance.

3.2. Firefly Algorithm

The firefly algorithm is one of the novel optimization algorithms based on swarm intelligence which was first introduced by X.Yang in 2008 [14]. It was inspired by the natural behavior of fireflies. The firefly algorithm randomly distributes a number of artificial fireflies in the search space at the beginning. All of the fireflies are unisexual and thus regardless of gender, each firefly can be attracted by any other firefly. Each firefly produces a light whose intensity depends on the optimality of its position and is proportional to its fitness value. The next step is comparing constantly the intensity of the light of each firefly with that of other fireflies and less bright fireflies moves towards brighter ones. Evidently, depending on the distance, fireflies receive lights with varying intensities; however, the brightest firefly moves randomly in search of space to increase its chance of finding the global optimum solution. Movement of the less bright firefly towards the brighter one is expressed through equation (1).

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

Where β_0 is the maximum coefficient of attraction between i_{th} and j_{th} fireflies, α is the coefficient of random displacement vector, γ is the light absorption coefficient for the environment, and r_{ij} is the Euclidean distance between two fireflies. Each firefly is compared to all others and if its fitness value is less than that of another one, it will be attracted according to equation (1). This trend continues to the last algorithm iteration when finally the optimum solution is obtained as the final solution. Main steps of the firefly algorithm can be expressed in the form of the pseudo-code briefed in **Algorithm 1** (FA).

Algorithm 1 Pseudo-code for FA main steps

```
Objective function f(x), x = (x_1, ..., x_d)^T
Generate initial population of fireflies x_i (i = 1, 2, ..., n)
Light intensity Ii at x_i is determined by f(x_i)
Define light absorption coefficient γ
while (t <MaxGeneration)
    for i = 1 : n all n fireflies
        for j = 1 : i all n fireflies
            if (I_j > I_i)
                Move firefly i towards j in d-dimension;
            end if
            Attractiveness varies with distance r via exp[−γr]
            Evaluate new solutions and update light intensity
        end for j
    end for i
    Rank the fireflies and find the current best
end while
Postprocess results and visualization
```

4 Multimodal Optimization

Constraints such as physical, temporal and economic limitations can prevent achievement of actual results; however, having knowledge of multimodal optimization solutions is very useful in engineering fields. In such cases, if multiple local and global solutions are available, the optimum system performance is obtained by switching between solutions. Since there are several solutions to many real-world problems, multimodal optimization algorithms are useful for solving these problems. Not only are these algorithms able to locate multiple optima in a single run, but they also preserve their population diversity. The reason why classic optimization techniques are not used to find multiple solutions shows their unreliability in finding more than one solution in multiple runs [15]. Evolutionary algorithms including Genetic Algorithms (GAs), Differential Evolution (DE), Particle Swarm Optimization (PSO), and Evolution Strategy (ES) are kinds of algorithms which has been tried to solve multimodal optimization problems. Referred algorithms [7, 10, 15-23] are among algorithms designed to the aforementioned criteria.

5 Multimodal Firefly Algorithm

Studies on multimodal optimization have mostly focused on the PSO and genetic algorithms. In this paper, like other meta-heuristic algorithms which have used unimodal algorithms for solving multimodal optimization problems, some changes are made on FA algorithm without the need for any additional parameter, and it has been utilized to solve multimodal optimization problems. In the proposed algorithm, the Coulomb's law in equation (2) has been used to calculate the electrostatic interaction between two fireflies. This technique was successfully used by J. Barrera and A. Coello in [3] to obtain a multimodal PSO algorithm. They have used this method to calculate forces between two particles which has also been used in the present paper to calculate the attraction between two fireflies.

$$F_{(j,i)} = \frac{1}{4\pi\varepsilon_0}.\frac{Q_i.Q_j}{r^2} \tag{2}$$

In this equation, $\frac{1}{4\pi\varepsilon_0}$ is the proportionality constant (Coulomb constant); Q_i and Q_j denote the magnitude of two charged particles, and r is the distance between two charges. According to this formula, force magnitude is proportional to magnitude of charges but it obeys inverse square law for distance. Hence, the attraction between two fireflies is calculated through the following equation.

$$\vec{F}_{(i,\vec{j})} = \alpha.\frac{f(p_i).f(p_{\vec{j}})}{\left\|p_i - p_j\right\|^2} \tag{3}$$

In this equation, α is equal to 1 and $f(p_i)$ is the fitness value for the firefly which will be attracted to one of the present fireflies. $f(p_{\vec{j}})$ is the vector of the fitness value of other fireflies to which the ith firefly is being compared. Finally, the destination firefly is obtained by using equation (4).

Algorithm 2 Pseudo-code for CFA main steps

```
Objective function f(x), x = (x_1, ..., x_d)^T
Generate initial population of fireflies x_i (i = 1, 2, ..., n)
Light intensity I_i at x_i is determined by f(x_i)
Define light absorption coefficient γ
while (t <MaxGeneration)
    γ= γ +t / (MaxGeneration*5);
    α = α * (1-t / )MaxGeneration*2);
    for i = 1 : n all n fireflies
        for j = 1 : i all n fireflies
            if (I_j > I_i)
                push(F,( I_i * I_j)/norm(x(i)-x(j))^2); */ in d-dimension
            end if
        end for j
        Move firefly i towards j^max(f) in d-dimension;
        Attractiveness varies with distance r via exp[−γr]
        Evaluate new solutions and update light intensity
    end for i
end while
Postprocess results and visualization
```

$$Fmax_{(i)} = \operatorname{argmax}_{\vec{F}} \vec{F}_{(i,j)} \qquad (4)$$

Calculating the above equation yields to the maximum value of $F_{(i,j)}$.Finally, the index of the j^th firefly, with the highest value of F, is calculated and equation (5) is obtained by changing equation (1).

$$x_i = x_i + \beta_0 e^{-\gamma r^2_{ij\,\mathrm{Fmax}(i)}} \left(x_{j\,\mathrm{Fmax}(i)} - x_i \right) + \alpha \left(rand - \frac{1}{2} \right) \qquad (5)$$

As a result, the i^th firefly is attracted to the firefly which has the highest value of F. Therefore, destination fireflies are not selected just based on the value of fitness value but from calculating the electrostatic interaction between other fireflies. This method prevents the attraction of other fireflies by the best firefly. Instead, fireflies are attracted by fireflies which besides having sufficient fitness value, must be at a close distance since distance is an effective parameter in their attraction. In each iteration in this case, each firefly compares its electrostatic interaction with others' and then moves toward the firefly which has the highest electrostatic interaction. As it was mentioned before, α is the coefficient of random displacement with a value considered to be [0.1- 1] at the beginning. This makes the firefly's movements to be random to some extent and to search for new sources; however, this value of α results in a less precise solution at the end of iteration. To prevent it, the coefficient of random movement is reduced in each iteration so as to reduce the randomness of the movement of the firefly toward the destination. Moreover, the value of γ is increased in each iteration so that fireflies take smaller steps at the end of the iteration. These two actions take place using equations (6) and (7).

$$\alpha = \alpha \times \left(1 - \frac{\text{Iteration}}{\text{MaxGeneration} \times 2} \right) \qquad (6)$$

$$\gamma = \gamma + \frac{\text{Iteration}}{\text{MaxGeneration} \times 5} \qquad (7)$$

Main steps of CFA can be summarized into the pseudo-code shown in Algorithm 2.

6 Experimental result

6.1. Test Functions

The experiments have been performed on benchmark functions common in multimodal optimization. Specifications of these algorithms are presented in Table (1).

6.2. Configurations

All algorithms were implemented in Matlab 2013 and were run in a computer equipped with an Intel Core(TM) i7-3632QM 2.2 GH processor and 8 gigabytes of RAM.

Table 1 Test Function

	Test function	Range	Peaks Global/Local
1	$f\left(x_1,x_2\right)=20-(-20.\exp\left(-0.2\sqrt{\frac{1}{2}\left(x_1{}^2+x_2{}^2\right)}\right)-e^{\left(\frac{1}{2}\left(\cos\left(2\pi x_1\right)+\cos\left(2\pi x_2\right)\right)+20+e^1\right)}$	$-5 < x_1 < 5$ $-5 < x_2 < 5$	**1/1**
2	$f\left(x_1,x_2\right)=100-\left(20+\left(x_1^2-10\cos\left(2\pi x_1^2\right)\right)+\left(x_2^2-10\cos\left(2\pi x_2^2\right)\right)\right)$	$-5.12 < x_1 < 5.12$ $-5.12 < x_2 < 5.12$	**1/1**
3	$f\left(x_1,x_2\right)=220-\sum_{i=1}^{5} i\ \cos(\left(i+1\right)x_1+i).\sum_{i=1}^{5} i\ \cos(\left(i+1\right)x_2+i).$	$-5.12 < x_1 < 5.12$ $-5.12 < x_2 < 5.12$	**4/201**
4	$f\left(x_1,x_2\right)=20-\left(\left(4-2.1x_1^2+\frac{x_1^4}{3}\right)x_1^2+x_1x_2+\left(-4+4x_2^2\right)x_2^2\right)$	$-1.9 < x_1 < 1.9$ $-1.1 < x_2 < 1.1$	**2/6**
5	$f\left(x_1,x_2\right)=2500-(\left(x_1^2+x_2-11\right)^2+\left(x_1+x_2^2-7\right)^2)$	$-6 < x_1 < 6$ $-6 < x_2 < 6$	**4/4**
6	$f\left(x_1\right)=5+sin^6\left(5\pi x\right)$	$0 < x_1 < 1$	**5/5**
7	$f\left(x_1\right)=5+e^{-2log\left(2\right)\times\left(\frac{x_1-0.1}{0.8}\right)^2}\times sin^6\left(5\pi x\right)$	$0 < x_1 < 1$	**1/5**

*f*1 = Ackley, *f*2 = Rastrigin, *f*3 = Shubert, *f*4 = Six-hump camel back, *f*5 = Himmelblau, *f*6 = Equal maxima, *f*7 = Decreasing maxima

6.3. Performance measures

To assess the performance of aforementioned algorithms in section (6.4), the following 7 criteria are considered and measured 50 runs.

1. Success Rates (SR): The percentage of performances in which all the optimum points have been found successfully.
   To calculate success rate, a user specified parameter, called the Level of Accuracy (LOA), is considered. This parameter is usually between (0,1] and is used to measure the difference between found solutions and the real optimum points in functions, so that if the difference between found solution and the real solution is less than the amount of LOA, then the found solution is counted as a successful solution [12].
2. Average Number Of Optima Found (ANOF): The average of optimum points found considering LOA for 50 runs
3. Global Average Number of Optima Found: The average of global optima found considering LOA for 50 runs
4. Average Function Evaluation is the average number of fitness function calling for 50 runs [1].
5. Success Performance (SP): This parameter is computable when the amount of SR is not zero [24]. SP is calculated from equation (8) :

$$SP=\frac{Avarage\ Number Of\ \ Function\ Evaluation\left(ANOF\right)}{SR} \tag{8}$$

.

   Based on the fact that algorithms with less ANOF and higher SR amount can be considered as a better one, it can be concluded that less SP amount is more acceptable.
6. Maximum Peak Ratio (MPR): The quality of optima is tested without considering the population distribution, and the performance metric, which is called the maximum peak ratio statistic (MPR), is adopted. The MPR is defined as follows:

$$MPR=\frac{\frac{1}{C}\sum_{i=1}^{C} f_i}{\frac{1}{q}\sum_{i=1}^{q} F_i} \tag{9}$$

   c: The number of found optimum point in the solution
   q: The number of real optimum points in the solution
   f_i: The quantity of fitness function obtained in the final population
   F_j: The quantity of real fitness value in objective function
7. Precision : the ratio of found optimum points to the number of real optimum points

$$\text{Precision} = \frac{c}{q} \tag{10}$$

6.4. Test and comparison results

The results of the experiments are shown in the tables 2-4 which have the accuracies of 10-1, 10-2 and 10-3 respectively. First columns of all three tables represent test functions; second columns are equivalent of implemented algorithms and the other columns are as written on top of each column. The best performance was reported in boldface. As it can be seen from the results, with the increase of the level of accuracy, the proposed algorithm has a better performance compared to FER-PSO [2], EPSO [3] and LS-FER-PSO [1] algorithms. It is shown in the tables that the presented algorithm and LS-FER-PSO algorithm have better performance compared to other algorithms. Comparing LS-FER-PSO algorithm and presented algorithm, it is shown that sometimes one performance is better than the other and vice versa. However it should be mentioned that the ratio of average function evaluation of LS-FER-PSO algorithm is 1.67 times higher than that of the presented algorithm. The performance of this algorithm proved its usefulness in solving optimization problems.

Fig. 1. shows the search space of f4. Fig. 2 also indicates the position of fireflies during the running of the proposed algorithm with 60 fireflies and 60 iterations using the f4 function. Fig. 2. A-D shows the position of particles in the 1^{st}, 10^{th}, 20^{th} and 30^{th} iterations. In fact, 4 out of the 6 available points were successfully found in 1800 function evaluation. However all optimum points can be found by increasing the number of fireflies. As it can be clearly observed in Fig. 2, by getting close to the end of the iterations, fireflies around optimum solutions become gradually more and more concentrated. Since the firefly algorithm was designed for maximum optimization problems, the average value of cost functions of fireflies increased in each iteration Fig. 3. Moreover, as seen in Fig. 4, as the concentration of fireflies around optima (i.e. around each other) increased, the standard deviation of cost functions decreased. The reason was that the more the process of the algorithm got closer to the end of the iteration, the more the fireflies and their cost functions got closer to form neighborhoods.

7 Conclusion

This paper proposed CFA multimodal firefly algorithm based on the Coulomb's law. This algorithm was successful in solving multimodal optimization problems. Results of experiments indicated that this unimodal optimization algorithm was successfully turned into a multimodal optimization algorithm through modifications. Two of the advantages of this algorithm are quickly yielding optimal results and not requiring additional parameter for being turned into a multimodal algorithm. According to the results, this algorithm can be considered as a reliable multimodal optimization algorithm.

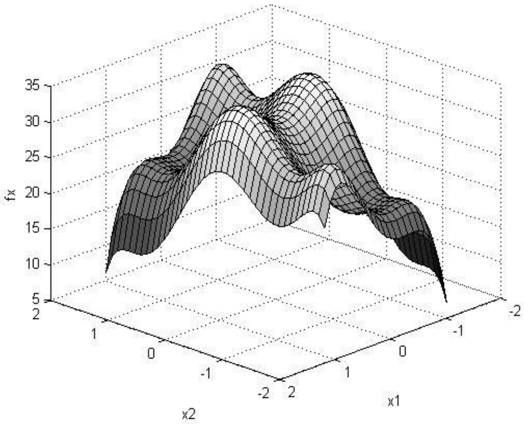


Fig. 1. search landscape of f4

Table2 The results of the experiments - accuracy: 10^{-1}

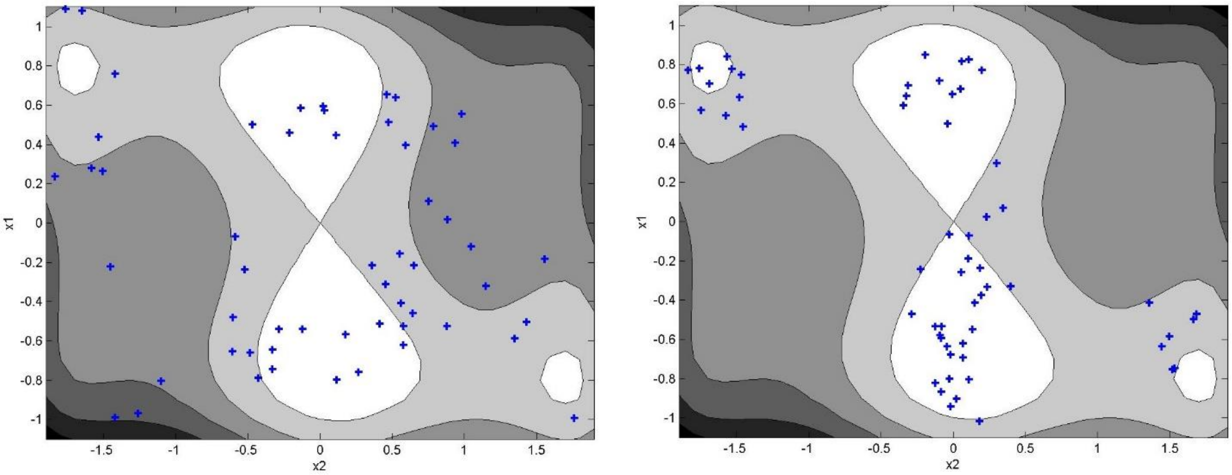
Function	Algorithm	Success Rate	Average Optima Found	Global Average Optima Found	Mean Peak Ration	Precision	Success Performance	Average Function Evaluation
Func-1	**CFA**	0	94.3	1	**1.013023**	**0.945670**	Inf	122000
	FER-PSO	0	37.38	1	0.94486	0.635588	Inf	102000
	EPSO	0	58.18	1	0.981776	0.664516	Inf	102000
	LS-FER-PSO	**0.08**	**108.68**	1	0.649761	0.908604	Inf	204000
Func-2	**CFA**	0	97.4	**1**	1.03153	0.944675	Inf	122000
	FER-PSO	0	58.9	0.98	1.235763	0.663146	Inf	102000
	EPSO	0	66.38	**1**	**0.996027**	0.766224	Inf	102000
	LS-FER-PSO	**0.52**	**110.3**	**1**	0.992886	**0.995518**	**927272.7**	204000
Func-3	**CFA**	0	147.34	**5.14**	**1.027689**	**0.962784**	Inf	170800
	FER-PSO	0	119.62	3.74	1.093445	0.805226	Inf	142800
	EPSO	0	62.18	3.02	1.040867	0.911363	Inf	142800
	LS-FER-PSO	0	**197.12**	4.2	0.190057	0.959919	Inf	285600
Func-4	**CFA**	0.14	4.56	2	**1.091253**	**0.984000**	87142.86	12200
	FER-PSO	0.02	4.2	2	1.135589	0.960333	510000	5100
	EPSO	0.12	4.7	2	1.022293	0.768690	**42500**	5100
	LS-FER-PSO	**0.98**	**5.98**	2	0.788084	0.976000	1020000	20400
Func-5	**CFA**	**1**	**4**	**4**	**0.888415**	0.456564	**12200**	12200
	FER-PSO	0.02	1.92	1.92	1.265238	0.03479	510000	5100
	EPSO	0.04	2.34	2.34	0.859798	0.037773	255000	5100
	LS-FER-PSO	**1**	**4**	**4**	2.205311	**0.466675**	20400	20400
Func-6	**CFA**	0.42	4.14	4.14	1.09713	0.855333	14523.80	6100
	FER-PSO	**1**	**5**	**5**	1.126117	0.648757	**5100**	5100
	EPSO	0	1	1	**1**	**1**	Inf	5100
	LS-FER-PSO	**1**	**5**	**5**	1.173182	0.868849	10200	10200
Func-7	**CFA**	0.12	3.14	1	**0.990534**	**1**	50833.33	6100
	FER-PSO	0.94	4.94	1	1.263032	0.812786	5425.532	5100
	EPSO	0	1	1	1.058738	**1**	Inf	5100
	LS-FER-PSO	**1**	**5**	1	1.144458	0.889643	**10200**	10200

Table3 The results of the experiments - accuracy: 10^{-2}

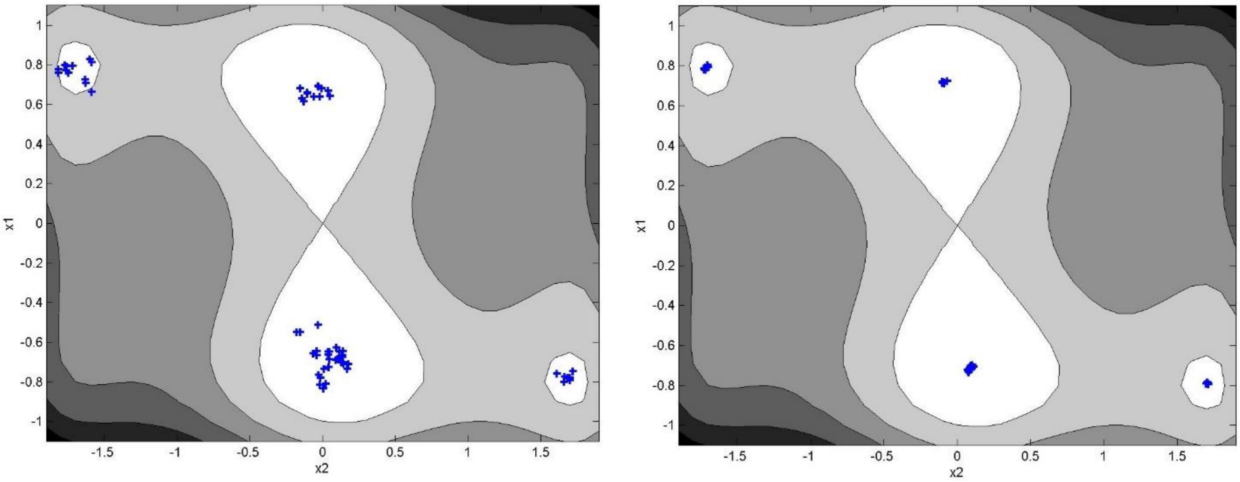
Function	Algorithm	Success Rate	Average Optima Found	Global Average Optima Found	Mean Peak Ration	Precision	Success Performance	Average Function Evaluation
Func-1	**CFA**	0	**53.8**	**1**	**1.020978**	**0.621089**	Inf	122000
	FER-PSO	0	6.38	0.64	0.630055	0.117527	Inf	102000
	EPSO	0	20.88	0.94	1.026456	0.248781	Inf	102000
	LS-FER-PSO	0	43.14	**1**	1.109575	0.616820	Inf	204000
Func-2	**CFA**	0	57.88	**1**	1.054625	0.562889	Inf	122000
	FER-PSO	0	7.64	0.24	1.245746	0.091978	Inf	102000
	EPSO	0	21.52	0.78	1.055243	0.247988	Inf	102000
	LS-FER-PSO	0	**114.34**	**1**	**1.000549**	**0.953799**	Inf	204000
Func-3	**CFA**	0	**85.08**	**5**	**1.057686**	**0.627016**	Inf	170800
	FER-PSO	0	10.88	0.68	0.92454	0.072021	Inf	142800
	EPSO	0	38.32	2.7	1.072754	0.580346	Inf	142800
	LS-FER-PSO	0	84.04	4	1.060134	0.603456	Inf	285600
Func-4	**CFA**	**0.06**	**4.46**	**2**	0.951791	**0.931333**	**203333.3**	12200
	FER-PSO	0	3.88	**2**	0.916055	0.890000	Inf	10200
	EPSO	0	3.44	1.76	**1.038658**	0.552857	Inf	10200
	LS-FER-PSO	0	4.02	**2**	1.173043	0.898333	Inf	20400
Func-5	**CFA**	0	**4**	**4**	**0.926575**	**0.408128**	**12200**	12200
	FER-PSO	0	0.46	0.46	0.846641	0.008104	Inf	10200
	EPSO	0	0.58	0.58	0.861226	0.018275	Inf	10200
	LS-FER-PSO	**1**	**4**	**4**	1.513459	0.404559	20400	20400
Func-6	**CFA**	0.02	3.02	3.02	0.866844	0.575548	305000	6100
	FER-PSO	**1**	**5**	**5**	0.995434	0.636006	**5100**	5100
	EPSO	0	1	1	**1**	**1**	Inf	5100
	LS-FER-PSO	**1**	**5**	**5**	1.159774	0.87873	10200	10200
Func-7	**CFA**	**1**	2.96	1	0.980653	0.796000	Inf	6100
	FER-PSO	**1**	**5**	1	0.824626	0.777659	**5100**	5100
	EPSO	0	1	1	**1.058738**	**1**	Inf	5100
	LS-FER-PSO	**1**	**5**	1	1.090589	0.92869	10200	10200

Table4 The results of the experiments - accuracy: 10^{-3}

Function	Algorithm	Success Rate	Average Optima Found	Global Average Optima Found	Mean Peak Ration	Precision	Success Performance	Average Function Evaluation
Func-1	**CFA**	0	**20.24**	0.86	**1.069871**	**0.228672**	Inf	122000
	FER-PSO	0	0.78	0.22	1.437689	0.014452	Inf	102000
	EPSO	0	8.06	0.5	1.103679	0.092075	Inf	102000
	LS-FER-PSO	0	13.22	**1**	1.123158	0.189530	Inf	204000
Func-2	**CFA**	0	18.1	0.62	1.030118	0.175755	Inf	122000
	FER-PSO	0	0.78	0	1.212728	0.009763	Inf	102000
	EPSO	0	8.44	0.42	0.986824	0.097171	Inf	102000
	LS-FER-PSO	0	**55.42**	**1**	**0.992886**	**0.461895**	Inf	204000
Func-3	**CFA**	0	**36.1**	**4.08**	**1.011276**	**0.266552**	Inf	170800
	FER-PSO	0	0.7	0.04	0.977142	0.004798	Inf	142800
	EPSO	0	16.04	1.08	1.03949	0.235603	Inf	142800
	LS-FER-PSO	0	24.14	4	1.039162	0.170459	Inf	285600
Func-4	**CFA**	**0**	**3.84**	1.88	**0.930002**	**0.845333**	Inf	12200
	FER-PSO	0	2.08	**2**	0.927387	0.470333	Inf	10200
	EPSO	0	1.28	0.78	0.785126	0.202500	Inf	10200
	LS-FER-PSO	0	3.14	**2**	1.182125	0.692333	Inf	20400
Func-5	**CFA**	0.9	3.88	3.88	0.798721	**0.412415**	13555.56	12200
	FER-PSO	0	0.24	0.24	1.702931	0.004307	Inf	10200
	EPSO	0	0.1	0.1	0.782353	0.001551	Inf	10200
	LS-FER-PSO	**1**	**4**	**4**	**1.054611**	0.170459	**20400**	20400
Func-6	**CFA**	0.06	2.76	2.76	1.29713	0.519389	101666.7	6100
	FER-PSO	**1**	**5**	**5**	1.012981	0.629383	**5100**	5100
	EPSO	0	1	1	**1**	**1**	Inf	5100
	LS-FER-PSO	**1**	**5**	**5**	1.208592	0.834246	10200	10200
Func-7	**CFA**	0	1.74	1	1.713465	0.528333	Inf	6100
	FER-PSO	0.96	4.96	1	1.368622	0.743413	**5312.5**	5100
	EPSO	0	1	1	**1.058738**	**1**	Inf	5100
	LS-FER-PSO	**1**	5	1	1.092660	0.922857	10200	10200



A: iteration 1 B: iteration 10



C: iteration 20

D: iteration 30

Fig. 2. the position of particles in different iterations (beginning to end)

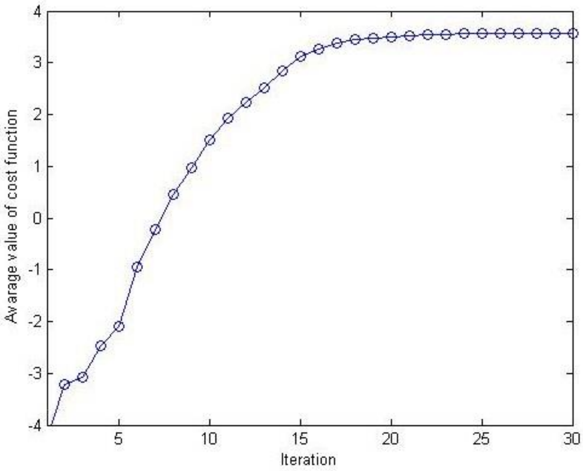


Fig. 3. The average value of cost function in each iteration

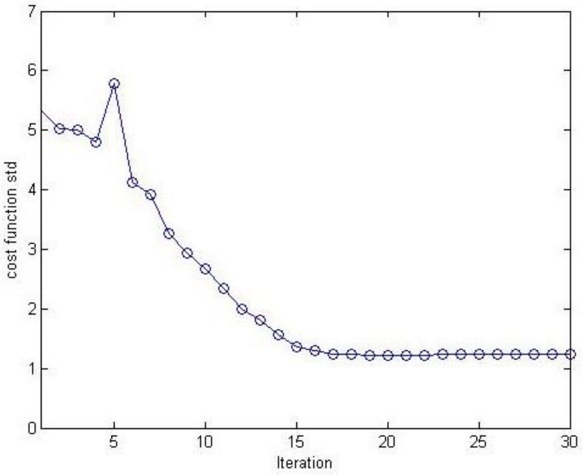


Fig. 4. The standard deviation of cost fucntions of fireflies in each iteration

References:

[1] B.-Y. Qu, J. J. Liang, and P. N. Suganthan, "Niching particle swarm optimization with local search for multi-modal optimization," *Information Sciences,* vol. 197, pp. 131-143, 2012.

[2] X. Li, "A multimodal particle swarm optimizer based on fitness Euclidean-distance ratio," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, 2007, pp. 78-85.

[3] J. Barrera and C. A. C. Coello, "A particle swarm optimization method for multimodal optimization based on electrostatic interaction," in *MICAI 2009: Advances in Artificial Intelligence*, ed: Springer, 2009, pp. 622-632.

[4] M. Li, D. Lin, and J. Kou, "A hybrid niching PSO enhanced with recombination-replacement crowding strategy for multimodal function optimization," *Applied Soft Computing,* vol. 12, pp. 975-987, 2012.

[5] E. Özcan and M. Yılmaz, "Particle swarms for multimodal optimization," in *Adaptive and Natural Computing Algorithms*, ed: Springer, 2007, pp. 366-375.

[6] T. Rahkar-Farshi, S. Behjat-Jamal, and M.-R. Feizi-Derakhshi, "An improved multimodal PSO method based on electrostatic interaction using n-nearest-neighbor local search," *arXiv preprint arXiv:1410.2056,* 2014.

[7] T. Grüninger and D. Wallace, "Multimodal optimization using genetic algorithms," *Master's thesis, Stuttgart University,* 1996.

[8] E. Dilettoso and N. Salerno, "A self-adaptive niching genetic algorithm for multimodal optimization of electromagnetic devices," *Magnetics, IEEE Transactions on,* vol. 42, pp. 1203-1206, 2006.

[9] R. K. Ursem, "Multinational GAs: Multimodal Optimization Techniques in Dynamic Environments," in *GECCO*, 2000, pp. 19-26.

[10] T. Rahkar-Farshi, O. Kesemen, and S. Behjat-Jamal, "Multi hyperbole detection on images using modified artificial bee colony (ABC) for multimodal function optimization," in *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*, 2014, pp. 894-898.

[11] X. Li, "Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization," in *Genetic and Evolutionary Computation–GECCO 2004*, 2004, pp. 105-116.

[12] X. Li, "Niching without niching parameters: particle swarm optimization using a ring topology," *Evolutionary Computation, IEEE Transactions on,* vol. 14, pp. 150-169, 2010.

[13] J. Zhang, J.-R. Zhang, and K. Li, "A sequential niching technique for particle swarm optimization," in *Advances in Intelligent Computing*, ed: Springer, 2005, pp. 390-399.

[14] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic algorithms: foundations and applications*, ed: Springer, 2009, pp. 169-178.

[15] S. W. Mahfoud, "Niching methods for genetic algorithms," *Urbana,* vol. 51, 1995.

[16] J.-H. Seo, C.-H. Im, C.-G. Heo, J.-K. Kim, H.-K. Jung, and C.-G. Lee, "Multimodal function optimization based on particle swarm optimization," *Magnetics, IEEE Transactions on,* vol. 42, pp. 1095-1098, 2006.

[17] M. Li and J. Kou, "Crowding with nearest neighbors replacement for multiple species niching and building blocks preservation in binary multimodal functions optimization," *Journal of Heuristics,* vol. 14, pp. 243-270, 2008.

[18] A. Passaro and A. Starita, "Particle swarm optimization for multimodal functions: a clustering approach," *Journal of Artificial Evolution and Applications,* vol. 2008, p. 8, 2008.

[19] K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *Evolutionary Computation, IEEE Transactions on,* vol. 8, pp. 211-224, 2004.

[20] K. D. Koper, M. E. Wysession, and D. A. Wiens, "Multimodal function optimization with a niching genetic algorithm: A seismological example," *Bulletin of the Seismological Society of America,* vol. 89, pp. 978-988, 1999.

[21] A. Anderson, C. McNaught, J. MacFie, I. Tring, P. Barker, and C. Mitchell, "Randomized clinical trial of multimodal optimization and standard perioperative surgical care," *British journal of surgery,* vol. 90, pp. 1497-1504, 2003.

[22] K. Parsopoulos and M. Vrahatis, "Modification of the particle swarm optimizer for locating all the global minima," *Artificial Neural Networks and Genetic Algorithms,* pp. 324-327, 2001.

[23] R. Brits, A. P. Engelbrecht, and F. van den Bergh, "Locating multiple optima using particle swarm optimization," *Applied Mathematics and Computation,* vol. 189, pp. 1859-1883, 2007.

[24] J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. Suganthan, C. A. C. Coello*, et al.*, "Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization," 2006.