A Gaussian Firefly Algorithm

Sh. M. Farahani, A. A. Abshouri, B. Nasiri and M. R. Meybodi

Abstract—**Firefly algorithm is one of the evolutionary optimization algorithms, and is inspired by fireflies behavior in nature. Each firefly movement is based on absorption of the other one. In this paper to stabilize firefly's movement, it is proposed a new behavior to direct fireflies movement to global best if there was no any better solution around them. In addition to increase convergence speed it is proposed to use Gaussian distribution to move all fireflies to global best in each iteration. Proposed algorithm was tested on five standard functions that have ever used for testing the static optimization algorithms. Experimental results show better performance and more accuracy than standard Firefly algorithm.**

Index Terms— **Algorithm, optimization, Global search, Local search.**

I. INTRODUCTION

The meaning of optimization is finding a parameter in a function that makes a better solution. All of suitable values are possible solutions and the best value is optimum solution [1]. Often to solve optimization problems, optimization algorithms are used. Classification of optimization algorithm can be carried out in many ways. A simple way is looking at the nature of the algorithms, and this divides the algorithms into two categories: deterministic algorithm, and stochastic algorithms. Deterministic algorithms follow a rigorous procedure, and its path and values of both design variables and the functions are repeatable. For stochastic algorithms, in general we have two types: heuristic and metaheuristic. Nature-inspired metaheuristic algorithms are becoming powerful in solving modern global optimization problems. All metaheuristic algorithms use certain tradeoff a randomization and local search [2], [3], [4].

Stochastic algorithms often have a deterministic component and a random component. The stochastic component can take many forms such as simple randomization by randomly sampling the search space or by random walks. Randomization provides a good way to move away from local search to the search on global scale. Most stochastic algorithms can be considered as metaheuristic and good examples are Genetic Algorithm (GA) [5], [6]. Many modern metaheuristic algorithms were developed based on the swarm intelligence in nature like PSO and AFSA [7], [8],

[9].

For example, Firefly algorithm is developed by the Xin-She Yang shows its superiority over some traditional algorithms [10], [11]. Firefly algorithm is inspired by fireflies in nature. Fireflies in nature are capable of producing light thanks to special photogenic organs situated very close to the body surface behind a window of translucent cuticle [12].

Firefly algorithm has some disadvantage such as trapping into several local optimums. Firefly algorithm do local search as well and sometimes can't get rid of them. Firefly algorithm parameters are set fixed and they do not change by the time. In addition Firefly algorithm does not memorize any history of better situation for each firefly and this causes they move regardless of it, and they miss their situations.

This paper aims to formulate a new Firefly algorithm and to provide the comparison study of the new-firefly with standard Firefly algorithm. The rest of this paper is organized as follows: it outlines the Firefly algorithm in section II, and then describes random walk is described in section III, new Firefly algorithm is introduced in section IV. Experimental settings and results are presented in section V. Section VI concludes the paper.

II. FIREFLY ALGORITHM

The Firefly algorithm was developed by Xin-She Yang [13], [14] and it is based on idealized behavior of the flashing characteristics of fireflies. For simplicity, we can summarize these flashing characteristics as the following three rules:

All fireflies are unisex, so that one firefly is 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. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If no one is brighter than a particular firefly, it will move randomly.

The brightness of a firefly is affected or determined by the landscape of the objective function to be optimized [15], [16].

Assume continuous optimization problem where the task is to minimize cost function $f(x)$ for $x \in S \subset R^n$ i.e. find x^* such as:

$$f(x^*) = min_{x \in S} f(x) \qquad (1)$$

For solving an optimization problem by Firefly algorithm iteratively, there is a swarm of m agents (fireflies) and x_i represents a solution for firefly i in whereas $f(x_i)$ denotes its cost.

Initially all fireflies are dislocated in S (randomly or employing some deterministic strategy). $S_k(k = 1, ..., d)$ In the d dimensions should be determined by the actual scales of

the problem of interest. For simplicity we can assume that the attractiveness of a firefly is determined by its brightness or light intensity which in turn is associated with the encoded objective function. In the simplest case for an optimization problem, the brightness I of a firefly at a particular position x can be chosen as $I(x)\alpha f(x)$. However, the attractiveness β is relative, it should vary with the distance r_{ij} between firefly i and firefly j. As light intensity decreases with the distance from its source and light is also absorbed in the media, so we should allow the attractiveness to vary with degree of absorption [19], [12].

The light intensity I(r) varies with distance r monotonically and exponentially. That is:

$$I = I_0 e^{-\gamma r} \qquad (2)$$

Where I_0 the original light intensity and γ is is the light absorption coefficient. As firefly attractiveness is proportional to the light intensity seen by adjacent fireflies, we can now define the attractiveness β of a firefly by Eq. (3) [17], [15].

$$\beta = \beta_0 e^{-\gamma r^2} \qquad (3)$$

Where r is the distance between each two fireflies and β_0 is their attractiveness at $r=0$ i.e. when two fireflies are found at the same point of search space S [12], [18]. In general $\beta_0 \in [0,1]$ should be used and two limiting cases can be defined: when $\beta_0 = 0$, that is only non-cooperative distributed random search is applied and when $\beta_0 = 1$ which is equivalent to the scheme of cooperative local search with the brightest firefly strongly determining other fireflies positions, especially in its neighborhood[13].

The value of γ determines the variation of attractiveness with increasing distance from communicated firefly. Using $\gamma=0$ corresponds to no variation or constant attractiveness and conversely setting $\gamma \to \infty$ results in attractiveness being close to zero which again is equivalent to the complete random search. In general $\gamma \in [0,10]$ could be suggested [13].

It is worth pointing out that the exponent γr can be replaced by other functions such as γr^m when m > 0. The distance between any two fireflies i and j at x_i and x_j can be Cartesian distance in Eq (4).

$$r_{ij} = \|x_i - x_j\|_2 = \sqrt{\sum_{k=1}^{d}(x_{i,k} - x_{j,k})^2} \qquad (4)$$

The firefly i movement is attracted to another more attractive (brighter) firefly j is determined by:

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2}(x_j - x_i) + \alpha \varepsilon_i \qquad (5)$$

Where the second term is due to the attraction, while the third term is randomization with the vector of random variable ε_i being drawn from a Gaussian distribution and ($\alpha \in [0,1)$) [11], [14]. In [15] a Lévy distribution is used instead of Gaussian one. Schematically, the Firefly algorithm can be summarized as the pseudo code in pseudo code 1.

1. Objective function f(x), x=$(x_1, x_2, \ldots, x_d)^{\mathrm{T}}$
2. Initialize a population of fireflies $x_i (i = 1,2,..,n)$
3. Define light absorption coefficient γ
4. **While** (t<MaxGeneration)
5.   **For** i=1:n   (all n fireflies)
6.   **For** j=1:i
7.   Light intensity I_i at x_i is determined by $f(x_i)$
8.   **If** $(I_i > I_j)$
9.   Move firefly i towards j in all d dimensions (Apply Eq. (5))
10.   **Else**
11.   Move firefly i randomly
12.   **End if**
13.   Attractiveness varies with distance r via $\exp[-\gamma r^2]$ $(\beta = \beta_0 e^{-\gamma r_{ij}^2})$
14.   Evaluate new solutions and update light intensity
15.   **End for j**
16.   **End for i**
17.   Rank the fireflies and find the current best
18. **End while**
19. Postprocess results and visualization.

Pseudo Code 1 Standard Firefly Algorithm

III. RANDOM WALK

A random walk is a process that consists of a series of the consecutive random step. From the view of the mathematic, if S_N shows the sum of the consecutive random step X_i then S_N forms a random step length that is shown in Eq. (6).

$$S_N = \sum_{i=1}^{N} X_i = X_1 + \cdots + X_N \qquad (6)$$

That X_i is a random walk that is drawn from a random distribution. This relation can be shown as a recursive equation like Eq. (7).

$$S_N = \sum_{i=1}^{N-1} + X_N = S_{N-1} + X_N \qquad (7)$$

In Eq. (7), the next value for S_N is just depends on the current value of the S_{N-1} and the movement of X_N from current state to next state.

Step length in a random walk can be fixed or variable. This movement can be in two directions front or back. Suppose a multi-dimensional movement that can occur in any directions, so a random walk can be defined as Eq. (8).

$$S_{t+1} = S_t + w_t \qquad (8)$$

That S_t is a current position or state in t and W_t is a step or a random variable form a distribution. If each step occurs in n dimensional space, a random movement is defined as Eq. (9).

$$S_N = \sum_{i=1}^{N} X_i \qquad (9)$$

That is a random walk in high dimension. In addition there isn't any reason to exist a fixed step length but step length can be according to a particular distribution. If this distribution is a Gaussian distribution, random walk is a Brawnian movement [13].

IV. PROPOSED ALGORITHM

In this section we propose a new firefly algorithm called GD-FF (Gaussian Distribution Firefly) algorithm. This new algorithm applies three behaviors to improve performance of firefly algorithm. The first behavior is an adaptive step length that changes random step length by the time and the other one is personal behavior or directed movement that directs random movement to towards global best. The last behavior is a social behavior that change the position of each fireflies based on a Gaussian distribution. In following sections we will define these behaviors.

A. Adaptive Step Length

In standard Firefly algorithm, firefly movement step length is a fixed value. So all the fireflies move with a fixed length in all iterations. Due to the fixed step length, the algorithm will miss better local search capabilities and sometimes it traps into several local optimums. It is better that Firefly algorithm do search in space globally in first iterations and in the end of iterations it exploit the particular place to extract better solutions. In proposed algorithm, it is defined a weight for α that depends on iterations and it always produce a value less than one. This coefficient is determined by Eq. (10).

B. Adaptive Step Length

$$W_{itr} = X + \frac{(itr_{max} - itr)^n}{(itr_{max})^n} + (Y - X) \quad (10)$$

Where n>=1. In Eq. (6), weight of W_{itr} is defined based on current iteration number and the last iteration number. Value of W_{itr} is between X and Y, and reduces by the time. Because α ∈ [0, 1], so X=0 and Y=1. n could be a linear or non-linear coefficient and itr_{max} is maximum number of iteration and itr is iteration i [18].

In Eq. (10) n parameter changes by dimension of each fireflies and its value determined by:

$$n = 10^{(-dimension)} \quad (11)$$

In Eq. (11) when dimension is high, n is a low value to produce a low value for W_{itr} too, and algorithm can search more accurate.

This strategy make an adaptive methods to change step length by the time and move fireflies with a long step in first iteration and make strong local search in final iteration.

C. Directed Movement

In addition in standard Firefly algorithm, firefly movement is based on light intensity and comparing it between each two fireflies. Thus for any two fireflies, the less bright one will move towards the brighter one. If no one is brighter than a particular firefly, it will move randomly. In proposed algorithm this random movement is directed, and that firefly moves towards best solution with better cost in that iteration. The firefly i movement is attracted to best solution that is more attractive (brighter). This causes that if there was no local best in each firefly's neighborhood; they move towards best solution and make better position for each firefly for next iteration and they will get more near to global best. Firefly's movement in this model is exactly similar to Eq. (5) in standard Firefly algorithm.

D. Social Behavior

Random walk is a random process which consists of taking a consecutive random step series of consecutive random steps. Here the step size or length in a random walk can be fixed or varying. If the step length obeys the Gaussian distribution, the random walk becomes the Brownian motion [13]. In standard Firefly algorithm, agents move by just a predefined movement that guides them to better position in their neighborhood. In order to move all fireflies in a same manner, it is used random walk concepts to move all of the agents based on a Gaussian distribution. In proposed algorithm, at the end of each iteration, it is introduced normal Gaussian distribution that is shown in Eq. (12).

$$p = f(x|\mu, \delta) = \frac{1}{\delta\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\delta^2}} \quad (12)$$

Where x is an error between best solution and fitness value of firefly i.

$$x = f(g_{best}) - f(x_i) \quad (13)$$

μ is mean and δ is standard deviation. Because of using standard normal distribution, it is set to μ=0 and δ=1. Then a random number will be drawn from this Gaussian distribution that is related to each firefly probability (p). Social behavior of fireflies is introduced by:

$$x_i = x_i + \alpha * (1 - p) * U(x, y) \quad (14)$$

That U(x, y) in Eq. (14) is a random number between [0,1]. In addition for social behavior But firefly i new position causes better cost, it will move to that new position. New firefly algorithm can be summarized as the pseudo code is shown in pseudo code 2. This strategy makes a social behavior for all fireflies and they move towards global best.

1. Objective function f(x), x=$(x_1, x_2, \ldots, x_d)^T$
2. Initialize a population of fireflies $x_i (i = 1, 2, \ldots, n)$
3. Define light absorption coefficient γ
4. **While** (t<MaxGeneration)
5.   **For** i=1:n   (all n fireflies)
6.   **For** j=1:i
7.   Light intensity is determined by f(x_i)
8.   **If** ($I_i > I_j$)
9.   Move firefly i towards j in all d dimensions
10.   **Else**
11.   Move firefly i towards best solution in that iteration
12.   **End if**
13.   Attractiveness varies with distance via $\exp(^{-\gamma r^2})$)
14.   **End for j**
15.   **End for i**
16.   Rank the fireflies and find the current best
17.   Define normal distribution
18.   **for** k = 1: n all n fireflies
19.   Draw a random number from defined distribution
20.   And apply Eq. (14).
21.   Evaluate new solution(new_cost(k))
22.   **If**((new_cost(k)<cost(i))&&(new_cost(k)< last_cost_iteration(k)))
23.   Move firefly i towards current best
24.   **End if**
25.   **End for** k
26. **End while**
27. Postprocess results and visualization

Pseudo code 2 GD-FF Algorithm

All of proposed behavior, direct fireflies direct to global best better and they shrinks to optimum solution. In GD-FF, random step is directed and is replaced by a movement that is more accurate than standard Firefly algorithm.

V. EXPERIMENTAL RESULTS

Performance of GD-FF algorithm is tested on a number of benchmark functions (table 1) which have been extensively used [19]. The benchmark functions include two unimodal functions, Rosenbrock and Sphere, and three multimodal functions, Rastrigin, Griewank and Ackley. The Rastrigin function has many local optima around the global optima and no correlation among its variables. The Ackely function is the only function, which introduces correlation between its variables. Table 1 shows the values that have been used for the dimension of these functions, the range of the corresponding initial position of the fireflies, and the goal for each function that has to be achieved by the algorithms [20], [21], [22], [23].

Table I Standard Test Functions

	Function	Range
F1	**Sphere**= $\sum_{i=1}^{n} x_i^2$	±100
F2	**Rosenbrock**= $\sum_{i=1}^{n-1}(100(x_{i+1} - x_i^2)^2 + (x_i-1)^2)$	±50
F3	**Rastrigin**=$\sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i) + 10)$	±5.12
F4	**Griewank**=$\sum_{i=1}^{n}\left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	±600
F5	**Ackley**= $20 + e - 20 * e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)}$	±32

In entire proposed algorithm population size is set to 30, and all of the fireflies are located in search space randomly. All of the results are in 10, 20 and 30 dimensions. The interval of this random search space is limited to the function that is applied for testing the proposed algorithm. Results are the mean fitness of the best value founded in 30 times separated run with independent seeds. In order to make a normal attractiveness, β_0 is set to 1 and for standard Firefly algorithm to do local search, γ is set to 1, initially and itr_{max} = 1000. In GD-FF algorithm the initial value of α is 0.7 and value of n changes by dimension of each fireflies and its value determines by Eq. (11).

Simulation results are shown in table II-VI and results of GD-FF is compared with standard Firefly algorithm and PSO and PSO-TVIW [24]. Table II shows better performance for GD-FF than other algorithms. Because of directing random movement GD-FF can get rid of trapping into several local optimums.

In standard Firefly algorithm when there is no any better solution in search space fireflies moves randomly and explore whole landscape, but sometimes this movement makes worse situation for them because they do not check their next position and move from current state to next state. So in GD-FF because each fireflies doesn't move except in a condition that their next state better than current state, so we can get sure they don't leave better state and go to worse one.

Also Gaussian distribution can produce accurate value of P

for each fireflies for example for a firefly that is far from global best P is a less value and it's normalization value (1-P) in Eq. (14) becomes high, therefore it moves by a long step length so can get more near to global best. By this method, away fireflies can shrinks to global best and locate in a better place for next iteration. Unlike for fireflies that are more near to global best, this step length is lower to get near to it. By this strategy GD-FF algorithm can direct all fireflies to better state than last state and it can get better result than other mentioned algorithms.

Adaptive step make a balance between exploration and exploitation property. In primary iteration GD-FF has long step length and explores search space and find better situation and in the last iteration it can exploit a special search space to extract better solution. As shown in table II GD-FF algorithm has better performance than standard Firefly algorithm that shows it can get rid of Firefly algorithm disadvantages.

Table II Average and standard deviation of best found result in 30 independent runs for F1 function.

Algorithm	Dimensions		
	10	20	30
Std.PSO	0.001	0.001	0.001
PSO-TVIW	0.001	0.001	0.001
Std.Firefly	1.45 (0.39)	4.23 (0.52)	7.08 (062)
GD-FF	**3.59e-056** **(1.96e-055)**	**3.96 e-035** **(2.63 e-035)**	**7.04 e -033** **(7.98 Ee-033)**

Table III Average and standard deviation of best found result in 30 independent runs for F2 function.

Algorithm	Dimensions		
	10	20	30
Std.PSO	21.71 (40.16)	52.21 (148.32)	77.61 (81.17)
PSO-TVIW	16.21 (14.98)	42.73 (72.61)	61.78 (48.93)
Std.Firefly	24.64 (3.76)	42.73 (72.61)	335.16 (35.46)
GD-FF	**5.55** **(0.99)**	**18.33** **(1.06)**	**29.62** **(1.76)**

Table IV Average and standard deviation of best found result in 30 independent runs for F3 function.

Algorithm	Dimensions		
	10	20	30
Std.PSO	2.33 (2.30)	13.81 (3.50)	6.65 (21.82)
PSO-TVIW	2.1184 (1.56)	16.36 (4.42)	24.346 (6.32)
Std.Firefly	4.80 (1.67)	11.49 (2.05)	26.13 (2.81)
GD-FF	**3.38** **(2.61)**	**5.87** **(4.29)**	**10.38** **(7.13)**

Table V Average and standard deviation of best found result in 30 independent runs for F4 function.

Algorithm	Dimensions		
	10	20	30
Std.PSO	0.16 (0.10)	0.25 (0.13)	0.0678 (0.24)
PSO-TVIW	0.092 (0.021)	0.1212 (0.52)	0.1486 (0.12)
Std.Firefly	1.04 (0.001)	1.56 (0.004)	1.93 (0.003)
GD-FF	**6.227e-008 (9.93e-008)**	**1.7199e-007 (1.94e-007)**	**1.5784e-006 (1.4680e-006)**

Table VI Average and standard deviation of best found result in 30 independent runs for F5 function.

Algorithm	Dimensions		
	10	20	30
Std.PSO	0.41 (1.42)	0.57 (3.10)	1.89 (2.21)
PSO-TVIW	0.238 (1.812)	0.318 (1.118)	0.63 (2.065)
Std.Firefly	1.75 (0.01)	1.78 (0.01)	1.81 (0.01)
GD-FF	**8.9410e-015 (3.08e-015)**	**3.11e-014 (6.77e-015)**	**1.3204e-014 (5.25e-015)**

In this paper, because of more enhanced performance of GD-FF than others, some plots of its operation are shown on test functions in Figs. 1-3. This figures show influence of GD-FF behavior on a best firefly to direct it to optimum value.
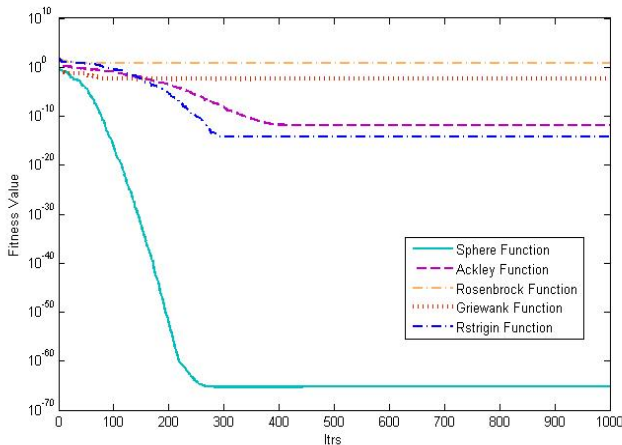

Figure 1 Comparison of simulation results of GD-FF in all functions in 10 dimensions.
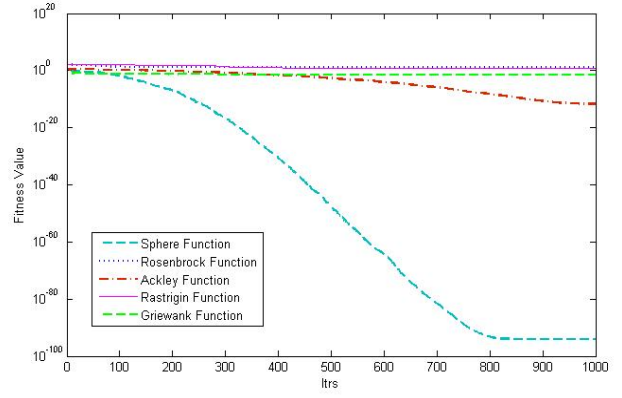

Figure 2 Comparison of simulation results of GD-FF in all functions in 20 dimensions.
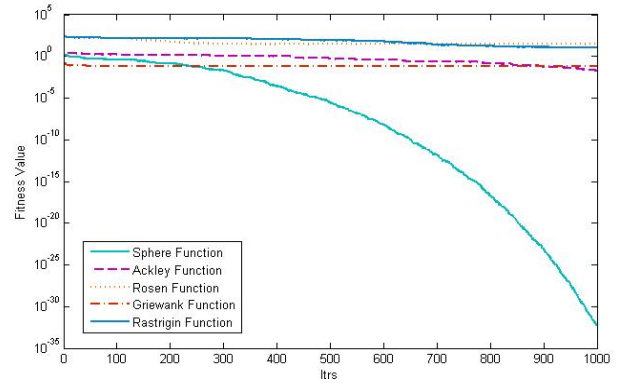

Figure 3 Comparison of simulation results of GD-FF in all functions in 30 dimensions.

As shown in Fig.1-3, GD-FF algorithm can get near to optimum solution in all test functions. But in sphere function it can perform better than the other functions because it doesn't have any local optimum and there is just a global optimum. Griewank function has local many local optimums and its figure is too hard so GD-FF can't get rid of local optimum as well. For comparison of the standard firefly algorithm and GD-FF algorithm, their performance is shown in Fig. 4 and Fig. 5 for two sphere and Ackley function in 30 dimensions.
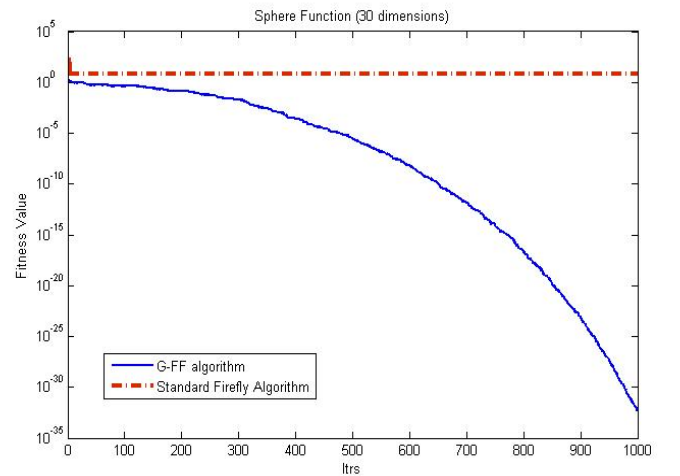

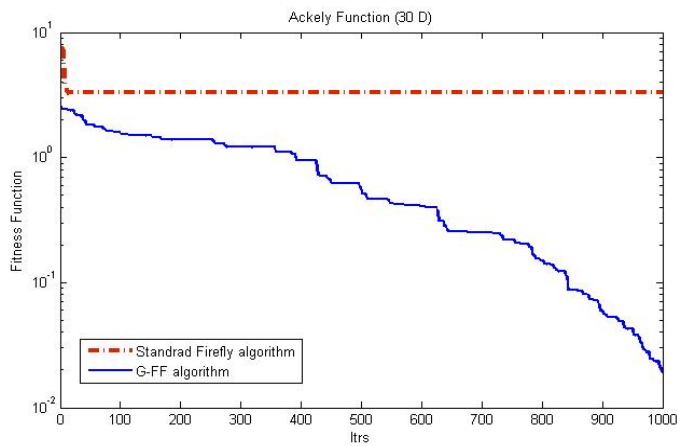Figure 4 Comparison of GD-FF and standard Firefly algorithm behavior for sphere function.

Figure 5 Comparison of GD-FF and standard Firefly algorithm behavior for Ackley function.

As shown in Fig. 4-5, standard Firefly algorithm traps into local optimum but GD-FF algorithm can get rid of it well.

VI. CONCLUSION

In this paper, three approaches are presented for improving standard Firefly algorithm. In first approach, initial value of the step length of movement is assumed to be big that this causes increasing speed of movement towards global optimum and prevent to trap into local optimum. After some iteration this parameter shrinks that causes focus on global optimum. By this method fireflies explore search space in primary iterations, and in the last iterations step length will get shorter so fireflies can exploit search a particular space and extract better solutions.

In proposed algorithm, if a firefly can't find any better firefly that is brighter in its neighborhood, it will move towards global best in that iteration so fireflies movement will direct to better solution and algorithm can guide them to better state, so they can get near to optimum solution at the end of iteration. Moving fireflies by a Gaussian distribution as a social behavior causes a better position for each of them for next iteration and fireflies with worse cost have more chance to move to global best with a longer step length. Simulation results show a better performance than standard Firefly algorithm.

REFERENCES

[1] Y. Liu and K. M. Passino, "Swarm Intelligence: A Survey", *International Conference of Swarm Intelligence*, 2005.

[2] T. Baeck, D. B. Fogel and Z. Michalewicz, "Handbook of Evolutionary Computation", Taylor & Francis, 1997.

[3] X. S. Yang, "Nature-Inspired Metaheuristic Algorithms", *Luniver Press,* 2008.

[4] X. S. Yang, "Engineering Optimization: An Introduction with Metaheuristic Applications", Wiley & Sons, New Jersey, 2010.

[5] I. H. Holland, "Adaptation in natural and Artifical Systems", University of Michigan, Press, Ann Abor, 1975.

[6] D. E. Goldberg, "Genetic Algorithms in Search, Optimisation and Machine Learning, Reading, Mass", Addison Wesley, 1989.

[7] J. Kennedy, R. C. Eberhart, "Particle swarm optimization", *IEEE International Conference on Neural Networks*, Piscataway, NJ., pp. 942-1948, 1995.

[8] J. Kennedy J., R. Eberhart and Y. Shi, "Swarm intelligence", Academic Press, 2001.

[9] L. X. Li, Z. J. Shao and J. X. Qian, "An optimizing Method based on Autonomous Animals: Fish Swarm Algoritm", System *Engineering Theory & Practice*, 2002.

[10] X. S. Yang, "Firefly algorithms for multimodal optimization", *Stochastic Algorithms:Foundations and Appplications (Eds O. Watanabe and T. eugmann), SAGA 2009, LectureNotes in Computer Science, 5792, Springer-Verlag, Berlin*, pp. 169-178, 2009.

[11] S. L. ukasik and AK. SÃlawomirZ, "Firefly algorithm for Continuous Constrained Optimization Tasks", 1st International Confernce on Computational Collective Intelligence, Semantic Web, Social Networks and Multiagent Systems, Springer-Verlag Berlin, Heidelberg, pp.169-178, 2009.

[12] B. G. Babu and M. Kannan, " Lightning bugs", Resonance, Vol. 7, No. 9, pp. 49-55, 2002.

[13] X. S. Yang, "Firefly algorithms for multimodal optimization", in Stochastic Algorithms Foundations and Applications, Stochastic Algorithms: Foundations and Applications (SAGA '09), Vol. 5792 of Lecture Notes in Computing Sciences, pp. 169-178, Spriger, October 2009.

[14] X. S. Yang, (2010). "Firefly Algorithm Stochastic Test Functions and Design Optimization". Int. *J. Bio-Inspired Computation*, vol.2, No. 2, pp.78-84, 2010.

[15] X.-S. Yang, "Firefly Algorithm, Lévy Flights and Global Optimization", *Research and Development in Intelligent Systems XXVI (Eds M. Bramer, R. Ellis, M. Petridis), Springer*, pp. 209-218, 2010.

[16] X. S. Yang, "Engineering Optimization: An Introduction with Metaheuristic Applications". Wiley & Sons, New Jersey, 2010.

[17] K. Krishnand, K, Ghose, and D, "Glowworms swarm based optimization algorithm for multimodal functions with collective robotics applications", Int. J. of Multiagent and Grid Systems, Vol. 2, No. 3, pp. 209-222, 2006.

[18] D. Yazdani, and M. R. Meybodi, "AFSA-LA: A New Model for Optimization", Proceedings of the 15th Annual CSI Computer Conference(CSICC'10), Feb. 20-22, 2010.

[19] S. Janson, M. Middendorf, " A hierarchical particle swarm optimizer and its adaptive variant", *IEEE Trans. on Systems*, *Man and Cybernetics*, Part B, Vol. 35, pp. 1272–1282, 2005.

[20] F. van den Bergh, A.P. Engelbrecht, "Effects of swarm size on cooperative particle swarm optimizers", in: Genetic and Evolutionary Computation Conference, San Francisco, CA, USA, 892–899, 2001.

[21] R. Mendes, J. Kennedy, J. Neves, "The fully informed particle swarm". simpler, maybe better", *IEEE Trans. On Evolutionary Computation* 8, pp. 204–210, 2004.

[22] D Bratton, J. Kennedy, "Defining a standard for particle swarm optimization", *IEEE Swarm Intelligence Symposium*, Honolulu, Hawaii, USA, pp. 120–127, 2004.

[23] R. Mendes, J. Kennedy, J. Neves, "Avoiding the pitfalls of local optima: how topologies can save the day", *12th Conference Intelligent Systems Application to Power Systems (ISAP2003),* Lemnos, Greece, 2003.

[24] Y. Li, A. Sun and H. T. Loh, "Adnaces of Computational Intelligence in Industrial Systems (ISAP2003), *IEEE World Congress on Computational Intelligence (International Joint Conference on Neural Networks),* Vol. 116, pp. 8834-8838, 2008.