A Modified Firefly Algorithm to solve Univariate Nonlinear Equations with Complex Roots

M.K.A. Ariyaratne#1, T.G.I. Fernando#2, S. Weerakoon*3

#Department of Computer Science, Faculty of Applied Sciences,
University of Sri Jayewardenepura, Gangodawila, Nugegoda, Sri Lanka
1anuradhaariyaratne@gmail.com
2gishantha@dscs.sjp.ac.lk

* Department of Mathematics, Faculty of Applied Sciences,
University of Sri Jayewardenepura, Gangodawila, Nugegoda, Sri Lanka
3sunethra.weerakoon@gmail.com

Abstract— **Recently developed meta-heuristic algorithms such as firefly algorithm, bat algorithm, particle swarm optimization and harmony search are now becoming popular for providing nearly accurate solutions for tough optimization problems. This paper addresses the problem of finding all roots of a given univariate nonlinear equation with real and complex roots using a modified firefly algorithm (MOD FA). The appropriate modifications are applied to the existing firefly algorithm (FA) by introducing an archive. Better fireflies are noted and stored in the archive during the iteration process and then their positions are replaced by new random ones. A comparison was carried out with the original firefly algorithm and also with the genetic algorithm (GA) which has a similar behaviour to the firefly algorithm. Computer simulations show that the proposed firefly algorithm performs well in solving nonlinear equations with real and complex roots within a specified region. The suggested method can be further extended to solve a given system of nonlinear equations.**

Keywords— **Firefly Algorithm; Nonlinear Equations; Archive; Real Roots; Complex Roots.**

I. INTRODUCTION

Solution of a single variable nonlinear equation can be defined as finding x, where $f(x) = 0$. Problems requiring the solutions of such nonlinear equations arise frequently in the fields of mathematics, engineering, physics and computer science. When thinking of a single nonlinear equation, solving $f(x) = 0$ is not easy, though it can be done in simple cases like finding roots of quadratic equations. If the function is complicated, approximations can be made using iterative procedures which are also known as numerical methods. Having their own drawbacks, none of these numerical approaches appear to be able to solve all types of nonlinear equations, especially when it has more than one solution. When the equation has complex solutions the situation is even worse. Most of the existing numerical approaches are associated with derivative of the function and thus solving equations with non-differentiable nonlinear functions like "Weierstrass function" is a challenge when the roots are needed in a specified region [1]. On the other hand, approaches like bisection method do not require derivative information but cannot be used in finding approximations although the given function is continuous on [a, b] but when not having opposite signs for $f(a)$ and $f(b)$ [2]. The Newton's method needs derivative information but fails in finding roots when multiple roots or very close roots exist in an equation [2]. The other main disadvantage of these numerical approaches is their inability to find more than one solution at a time, in a specified region. As such, it is clear that finding better algorithms to determine more than one root or all available roots of a function, simultaneously, without having to use its derivative information has become the need of the hour.

The remarkable performance of nature inspired algorithms over other classical optimization techniques encourages researchers to apply them for various difficult optimization problems. Recently developed algorithms like firefly algorithm, bat algorithm, cuckoo search and artificial bees' colony have proved their success over many difficult problems [3, 4, 5, 6, 13].

Firefly algorithm (FA) is one of the nature-inspired meta-heuristic algorithms developed by Xin She Yang, originally designed to solve continuous optimization problems [7]. It is capable of giving several possible approximations for a given problem rather than giving one globally best solution as in Bat Algorithm or Particle Swarm Optimization (PSO) algorithm. Since our problem of interest is also associated with getting more than one optimal solution in a specified region, we have selected the firefly algorithm. Our problem of interest can be defined as follows.

Let f be a function $s.t. f: D \rightarrow R$ where $D \subset C$. Neither the differentiability nor the continuity of f is required. The problem is to find all $x \in D$ $s.t. f(x) = 0$. Since the problem is handled as an optimization problem, the problem becomes finding $x \in D$ $s.t. |f(x)| = 0$. However it should be emphasized here that, since the function $f(x)$ may have multiple roots, the optimization problem $|f(x)| = 0$, also will have multiple optimal solutions. Therefore our objective turns out to be finding all such optimal solutions.

In this research, modifications to the original firefly algorithm are introduced to solve nonlinear equations with more than one real and/or complex root within a specified region. The modifications have been done by applying a similar concept to the elitism in Genetic Algorithms (GAs). In an iteration, the better fireflies who serve as solutions are selected and they are put into an archive while replacing their positions with random fireflies. The iterative process of the original firefly algorithm is also changed using a "flag." Single variable nonlinear equations that contain real and/or complex roots within a specified region were tested with the algorithm to make conclusions.

II. FIREFLY ALGORITHM

The original algorithm proposed by Yang is inspired by the flashing behaviour of fireflies. It relies on the following three idealized rules [7].

a. Fireflies are unisex so their attraction to each other is gender independent.
b. Attractiveness is proportional to their brightness, and for any two fireflies, the less brighter one is attracted by (and thus moves toward) the brighter one; however, the brightness can decrease as their distance increases; If there is no brighter one than a particular firefly, it moves randomly.
c. The brightness of a firefly is determined by the value of the objective function.

Algorithm 1 : Original Firefly Algorithm

Begin;
Initialize algorithm parameters:
MaxGen: the maximum number of generations
γ*: the light absorption coefficient*
r: the particular distance between two fireflies
D: the domain space

Define the objective function $f(X)$, *where* $X = (x_1, \ldots, x_d)^T$
Generate the initial population of fireflies, X_i $(i = 1,2, \ldots, n)$
Determine the light intensity I_i *of* i^{th} *firefly* X_i *via* $f(X_i)$

while $t < MaxGen$ *do*
  for $i = 1 : n$ *(all n fireflies) do*
    for $j = 1 : n$ *(all n fireflies) do*
      if $I_j > I_i$ *then*
        Move firefly i towards j by using eq (1);
      end if
    Attractiveness varies with distance r via $e^{-\gamma r^2}$ *using eq (2);*
    Evaluate new solutions and update light intensity;
    end for
  end for
  Rank the fireflies and find the current best;
end while
  Post process results and visualization;
End;

The initial population can be defined randomly with a set of feasible solutions for the problem. Then each firefly's light intensity is calculated using the problem specific objective function. Then each firefly in the population starts moving towards brighter fireflies according to the following equation.

$$x(i) = x(i) + \beta\big(x(j) - x(i)\big) + \alpha(rand - 0.5); \rightarrow (1)$$

$$where\ \beta = \beta_0\, e^{-\gamma r^2} \rightarrow (2),$$

β_0 *is the attraction at* $r = 0$;

The second term of the $\boldsymbol{Eqn(1)}$ is due to the attraction between two fireflies and the third term is a randomization term where α is the randomization factor drawn from the Gaussian or the uniform distribution.

Yang has proved the original algorithm's performance is relatively high when compared with Genetic Algorithms and Particle Swarm Optimization algorithms [7]. Popular two dimensional optimization problems were used in his original implementation to prove the idea. Because of its success we have focused on using the firefly algorithm to solve single variable nonlinear equations having both real and complex roots within a given region.

III. MODIFIED FIREFLY ALGORITHM

The two cornerstones of problem solving by search are exploration and exploitation. Nature inspired algorithms provide unique ways to implement exploration and exploitation over the search space [8]. Firefly algorithm also has its own ways of acquiring those; for example the movement of a firefly towards a brighter firefly using $\boldsymbol{Eqn\ (1)}$, has exploitation, because it exploits the solution of previous iteration to build the new solution and also it consists of a randomization factor for the exploration. Also the light absorption coefficient plays a great role in providing exploration. $\boldsymbol{\gamma}$, the light absorption coefficient can vary from 1 to Infinity, if the absorption is high then the attractiveness becomes low and the search becomes more random. It allows the search to go in a new direction rather than going towards the currently existing best solution. This factor is very important when the problem has more than one optimum solution to seek.

We have implemented the original firefly algorithm to the problem of solving nonlinear equations with several roots. It was successful for the functions having up to three roots but when more roots exist, parameter tuning and increase of population size together with number of iterations did not work. Thus we experimented on modifying the original algorithm and as a result the following modified algorithm has been developed.

Algorithm 2 : Modified Firefly Algorithm

Begin;
Initialize algorithm parameters:
MaxGen: the maximum number of generations
γ*: the light absorption coefficient*
r: the particular distance between two fireflies

Define the objective function $f(X)$;
Generate the initial population of fireflies, X_i $(i = 1,2, \ldots, n)$
Determine the light intensity I_i *of* i^{th} *firefly* X_i *via* $f(X_i)$

```
while t < MaxGen do
flag = true;
while flag = true and time < MaxGen do
  for i = 1 : n (all n fireflies) do
    for j = 1 : n (all n fireflies) do
      if I_j > I_i then
        Move firefly i towards j by using eq (1);
      end if
      Attractiveness varies with distance r via
                         e^{-γr^2} using eq (2);
      Evaluate new solutions and update light intensity;
    end for
  end for

  Find the fireflies with the eligibility criteria
                              |f(X)| < 0:001;
  Put them into the archive and replace the positions
                              with random fireflies;

  if no fireflies matching with eligibility criteria
    flag = false;
  end if

  if flag = false
    count = random integer between 0 and n;
    Create random fireflies up to count and
                       replace the population;
  end if

end while
end while
Post process results and visualization;
End;
```

The problem of root finding is viewed as an optimization problem. The aim is to seek for better fireflies whose $|f(x)|$~0, (10^{-3}).

A. Representation of a firefly

A firefly is represented as a complex number within a given region. For example a feasible firefly can be defined as

$$\{(x+yi) \in \mathbb{C} \mid x \in [-2,2], y \in [-2,2]\}$$

$-1.234+0.539i$ is one such firefly.

B. Distance between two fireflies (r)

To calculate the distance between two fireflies, we used the distance between the two complex numbers that represent those two fireflies.

C. Objective function

The absolute value $|f(x)|$ of the original function $f(x)$ is used as the objective function.

D. Light intensity of a firefly

Light intensity has a relation with the objective function. As in the original implementation of the firefly algorithm, the value of the objective function is used for the intensity of a firefly.

E. Brightness (β)

The brightness of a firefly is calculated using ***Eqn* (2)**.

F. Light absorption (γ)

Light absorption factor determines how much light is absorbed by the environment. When $\gamma = 0$, the attractiveness of a firefly is equal to its initial brightness. Since we need a lot of randomness in our search to find many different optimum values, we have kept it around 75.

G. Movement of a firefly

A movement of a firefly to a brighter firefly is determined by the ***Eqn* (1)**.

H. Archive and Flag

These are the two properties introduced in the new modification. Here an archive and a flag are used in order to improve the performance in finding all real and/or complex roots. So the new firefly algorithm works as follows.

After an iteration, we seek for fireflies satisfying$|f(x)| < 0.001$. Such fireflies are taken into the archive and their positions are replaced by random fireflies. A flag which is true at the beginning of each iteration is used to check poorly performed iterations. In such iterations we change the firefly population with a new random firefly population. After completing the run, fireflies in the archive are considered as the approximations for the roots of the nonlinear equation.

IV. Preliminary Numerical Examples

To illustrate the performance of the proposed modification we have selected some representative numerical examples from various categories of nonlinear functions.

1. The following one dimensional trigonometric equation (adapted from Goldberg and Richardson, 1987 [9]) has 51 real roots within the given interval and the modified firefly algorithm was capable of finding all of them (see Table 4, Figure 2 graph (a)). We have selected this to prove the ability of our approach in finding all the real roots of a nonlinear equation within a given interval.

$$y = \sin^3(5\pi x) \quad where\ x \in [-5,5]$$

2. The Weierstrass functions possesses the property of being continuous everywhere but differentiable nowhere [10]. Since most of the numerical approaches need to evaluate derivatives, it is difficult to employ a numerical approach to find roots of such a function. We used our method to find roots of the following Weierstrass function within the domain [-20, 20] which has 25 real roots (see Table 6, Figure 2 graph (b)).

$$W(x) = \sum_{i=1}^{3}(1/2^i)\sin(2^i x)$$

3. $y = \tan(x)$ is a popular trigonometric function with discontinuities. It has 25 real roots within [-40, 40]. In spite of having discontinuities between the given intervals, our approach gave approximations for all of them (see Table 5, Figure 2 graph (c)).

4. Popular numerical approaches like the Newton's method and the secant method converge more slowly for multiple roots than for the case of a simple root [11]. We have solved the following nonlinear equation with multiple roots and found that our approach is successful (see Figure 2 graph (e)).

$$y = x^4 - 2x^2 + 1, [-2, 2]$$

5. The following parabolic function has 6 roots. Since the derivative at 0 is equal to zero, the Newton's method cannot be applied to approximate the roots at zero. But the modified firefly algorithm is capable of finding them all (see Figure 2 graph (d)).

$$y = sin\,(x) + 1, [-20,20]$$

6. The suggested method is capable of finding complex roots. Computed results shown in Table 1, demonstrate the ability of the method in finding complex roots.

	Equation	Region	#of Roots
1	$y = x^2 + 1$	[-1.5,1.5] X [-1.5, 1.5]	2 complex
2	$y = x^3 + 2x^2 + 3x + 4$	[-2, 0] X [-2, 0]	2 Complex, 1 Real
3	$y = x^5 - 3x^4 + 3x^3 - 2x^2 - 5$	[-1, 3] X [-1, 3]	4 Complex, 1 Real
4	$y = x^7 - x^6 + 2x^5 - 3x^4 + 3x^3 - 2x^2 - 5$	[-1, 2] X [-1, 2]	6 Complex, 1 Real
5	$y = x^{10} - 3x^9 + x^8 - 7x^7 + x^6 - x^4 + 2x^2 - 5$	[-1, 4] X [-1, 4]	8 Complex, 2 Real
6	$y = x^{12} - 6x^{11} + x^{10} - 5$	[-1, 6] X [-1, 6]	10 complex, 2 Real
7	$y = x^{13} - 2x^{12} + 1$	[-1, 2] X [-1, 2]	10 Complex, 3 Real

Table 1: Nonlinear equations with complex roots

Region of the function is the area we seek for the roots. According to the notation we adopted, [-1, 2] X [-1, 2] region describes the area surrounded by the [-1, 2] real axis and the [-1, 2] imaginary axis.

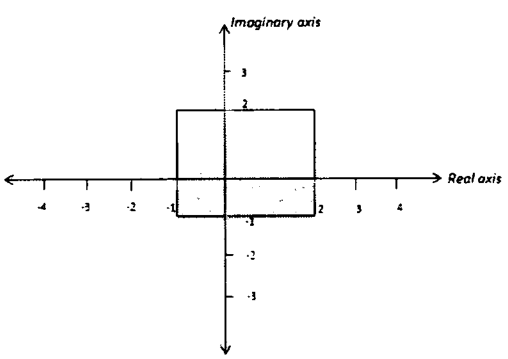

Fig. 1: [-1, 2] X [-1, 2] Region

V. Results and Discussions

The proposed firefly algorithm (MOD FA), the original firefly algorithm (FA) and the genetic algorithms (GA) are implemented to test the performances of the system. Since our problem of interest has many optimums, techniques like particle swarm optimization, bat algorithms which seek only one global optimum are not suitable for the situation. Genetic algorithm has a similar behaviour to the firefly algorithm and also a well-known optimization algorithm in the field [12]. We have also done the appropriate modifications to the original genetic algorithm (MOD GA) in order to compare the performance of the modified firefly algorithm. When all roots are real, the population size of 200 and number of iterations per run of 200 were used to measure the performance. 100 such runs were carried out. Results are formatted as; average number of roots found, (maximum number of roots found / total roots)*100%, so **19, (49%)** means the average number of roots found in 100 runs is 19 and the (maximum number of roots found in a run / total roots)*100% is 49%.

Function	GA	Modified GA	FA	Modified FA
$y = \sin^3(5\pi x)$	0, (0%)	51, (100%)	19, (49%)	51, (100%)
$y = \sum_{i=1}^{3} \left(\frac{1}{2^i}\right) sin\,(2^i\,x)$	1, (4%)	25, (100%)	13, (68%)	25, (100%)
$y = \tan\,(x)$	1, (4%)	25, (100%)	21, (96%)	25, (100%)
$y = x^4 - 2x^2 + 1$	2, (50%)	4, (100%)	4, (100%)	4, (100%)
$y = \sin(x) + 1$	1, (17%)	6, (100%)	6, (100%)	6, (100%)

Table 2: Performance of the algorithms for real roots over 100 runs

For the real roots situations, modified GA and modified FA performed well. The archiving property and diversifying the population during iterations is the reason for the good performance. When the nonlinear equations have complex roots, to optimize the performance we have to set the population size to 600 and the number of iterations per run to 600.

Equation	GA	Modified GA	FA	Modified FA
Table 1: Equation 1	0, (0%)	2, (100%)	2, (100%)	2, (100%)
Table 1: Equation 2	1, (33%)	1, (33%)	3, (100%)	3, (100%)
Table 1: Equation 3	1, (20%)	1, (20%)	5, (100%)	5, (100%)
Table 1: Equation 4	1, (14%)	1, (14%)	7, (100%)	7, (100%)
Table 1: Equation 5	1, (10%)	1, (10%)	9, (90%)	10, (100%)
Table 1: Equation 6	1, (8%)	3, (25%)	11, (92%)	12, (100%)
Table 1: Equation 7	1, (8%)	2, (15%)	8, (92%)	13, (100%)

Table 3: Performance of the algorithms for real and complex roots over 100 runs

Although the modified GA performed well in solving nonlinear equations with real roots situations, it was unable to find complex roots under given parameter settings. Experiments done with larger population sizes (1200, 1500) were able to give complex roots to some extent. To solve for

complex roots, modified GA needs a large chromosome population under the given parameter setting to provide approximations. But under the same conditions, modified FA was able to provide all approximations and according to the results obtained. Even the original firefly algorithm is successful to some extent.
The following tables show the approximations given by the modified firefly algorithm for the given problems.

$y = sin^3(5\pi x)$					
-5.000	-4.790	-4.610	-4.410	-4.190	-4.000
-3.800	-3.600	-3.400	-3.200	-3.000	-2.800
-2.600	-2.400	-2.200	-2.000	-1.800	-1.600
-1.400	-1.200	-1.000	-0.800	-0.600	-0.400
-0.200	0.000	0.200	0.400	0.600	0.800
1.000	1.2000	1.400	1.600	1.800	2.000
2.200	2.400	2.600	2.800	3.000	3.200
3.400	3.600	3.799	4.000	4.199	4.390
4.600	4.799	5.000			

Table 4: 51 roots given by the modified firefly algorithm for $y = sin^3(5\pi x)$

$y = tan(x)$				
0.000	3.091	6.307	9.460	12.552
15.640	18.920	21.950	25.105	28.320
31.410	34.560	37.710	-3.091	-6.307
-9.460	-12.552	-15.640	-18.920	-21.950
-25.105	-28.320	-31.410	-34.560	-37.710

Table 5: 25 roots given by the modified firefly algorithm for $y = tan(x)$

$y = sin(x) + \left(\frac{1}{2}\right) sin(2x) + \left(\frac{1}{4}\right) sin(4x) + \left(\frac{1}{8}\right) sin(8x)$				
-18.8500	-16.2600	-15.7100	-15.1500	-12.5700
-9.9800	-9.4300	-8.8700	-6.2800	-3.7000
-3.1400	-2.5900	0	2.5900	3.1400
3.7000	6.2800	8.8700	9.4300	9.9800
12.5700	15.1500	15.7100	16.2600	18.8500

Table 6: 25 roots given by the modified firefly algorithm for $y = \sum_{i=1}^{3} \left(\frac{1}{2^i}\right) sin\,(2^i x)$

$y = x^{13} - 2x^{12} + 1$				
1	1.99976	−0.91471	0.44213+ 0.84473i	0.44213 - 0.84473i
−0.48199 +0.78756i	−0.48199- 0.78756i	0.83087+ 0.51909i	0.83087- 0.51909i	−0.03356 +0.9344i
−0.03356- 0.93447i	−0.79997 +0.44787i	-0.79997- 0.44787i		

Table 7: 13 roots given by the modified firefly algorithm for $y = x^{13} - 12x^{12} + 1$

The following figure graphically represents the real roots found by the modified firefly algorithm.

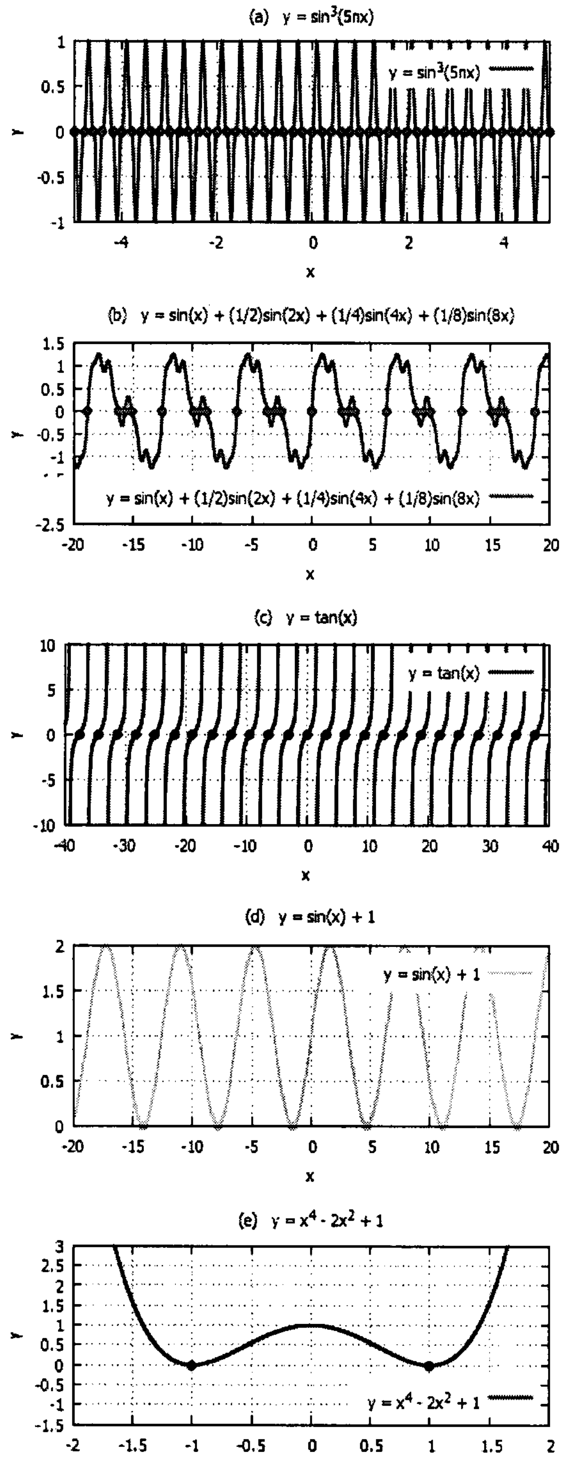

Fig. 2: Graphical representation of the roots found by the modified FA for the examples 1, 2, 3, 4, 5 in section IV

Since our aim is to find almost all the real/ complex roots in a given region, within a single run, we have tested the performance of the algorithms over a single run. The following graphs summarize the results of the convergence of the algorithms for $y = sin^3(5\pi x)$ and $y = x^{13} - 2x^{12} + 1$ within a single run (over 500 iterations).

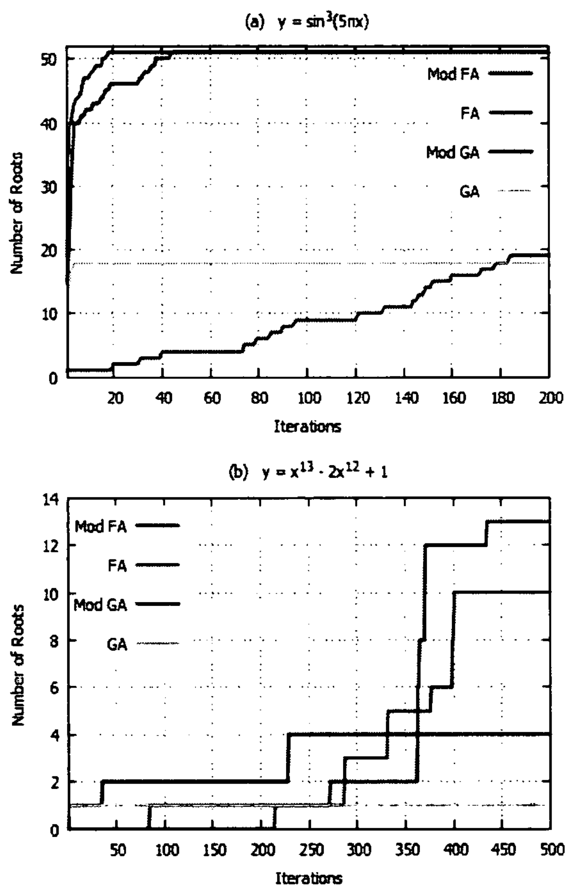

Fig. 3: Convergence ability of Mod FA, FA, Mod GA and GA within a single run for (a): $f(x) = \sin^3(5\pi x)$, (b): $f(x) = x^{13} - 12x^{12} + 1$

We have also measured the average CPU time used by the four algorithms solving real and complex functions. The following tables summarize the results.

Function	GA	Modified GA	FA	Modified FA
$y = \sin^3(5\pi x)$	2.1	7	4.1	18.8
$y = \sum_{i=1}^{3}\left(\frac{1}{2^i}\right)\sin(2^i x)$	2.6	7.8	4.2	4.7
$y = \tan(x)$	2.1	5.4	3.4	4.6
$y = x^4 - 2x^2 + 1$	2.3	11.4	4.3	14.1
$y = \sin(x) + 1$	2.1	6.6	4.0	8.3

Table 8: Computational Efficiency: average CPU time in (seconds) for single run, Real roots

Equation	GA	Modified GA	FA	Modified FA
Table 1: *Equation* 1	9.4	18	120	150
Table 1: *Equation* 2	23	28	266	240
Table 1: *Equation* 3	11	24	216	250
Table 1: *Equation* 4	12	24	118	284
Table 1: *Equation* 5	12	21	120	299
Table 1: *Equation* 6	11	19	170	190
Table 1: *Equation* 7	10	30	122	214

Table 9: Computational Efficiency: average CPU time in (seconds) for single run, Real and Complex roots

It is clear that for some situations modified GA performs faster than Modified FA. Though the Table 9 shows that GA, MOD GA & FA are more efficient than MOD FA, the graphs show that MOD FA is capable of finding more roots (or all the roots) than all the other algorithms. That is the very reason for taking more time.

VI. CONCLUSIONS

In this paper we present a modification to the existing firefly algorithm to solve nonlinear equations with several real and complex roots within a predefined interval. The modification includes an archive to collect best fireflies during iterations and replacing their positions with random ones. Also a flag was used to identify poor iterations and to change the randomness of the existing population. The simulation results for finding roots of several numerical examples including an application of Weierstrass function and complex polynomials suggest that this new firefly algorithm with the archiving property is capable of finding almost all real and complex roots of a nonlinear equation simultaneously with the given accuracy of 10^{-3}. It works for the multiple roots situations too. Comparison with other similar nature inspired algorithms including the original firefly algorithm clearly shows that this modified firefly algorithm outperforms all of them. This evidence suggests that the proposed firefly algorithm is by far the best performer in solving nonlinear equations with several real and complex roots.

With the results obtained, we can conclude that the proposed firefly algorithm is capable of giving reasonably good approximations for the nonlinear equations:

a. with several roots.
b. with multiple roots.
c. which are continuous but not differentiable.
d. which have discontinuities within the interval.
e. which are having a pattern within the given range.
f. which have roots over a large interval.
g. having complex roots as well as real roots in a given interval.

Accuracy of the roots found by the modified FA is within 10^{-3} tolerance. For higher accuracies one can treat these solutions as initial guesses and try out a suitable numerical approach. The accuracy of the solutions are limited to 10^{-3} because our objective here is to find almost all the solutions within the given region. The approximations provided here highly depend on the population size, number of iterations and also on the algorithm specific parameter values. It is essential to define the number of iterations and the population size properly according to the number of roots within the specified interval.

Differentiability and continuity of the nonlinear functions are inessential when applying nature inspired algorithms to obtain roots; thus they could be applied to the functions arising from various practical situations where it is impossible to apply formal numerical schemes. This can be considered as the biggest advantages of using nature inspired algorithms.

The basic firefly algorithm introduced by Yang is powerful, but the problem of finding all real and complex roots of a given nonlinear equation simultaneously has not been addressed before. Thus our approach of introducing an archive is undoubtedly advantageous. But still this approach needs higher number of iterations and a large firefly population when we need higher accuracies in approximations. To improve the proposed algorithm we can do a strong parameter analysis and fine tuning with an intention of reducing the population size. As an advancement to the suggested method, one can check its ability of solving a given system of nonlinear equations.

ACKNOWLEDGMENT

The authors would like to thank Dr. Xin She Yang for his valuable suggestions and guidance on the work.

REFERENCES

[1] S. Weerakoon, T.G.I. Fernando, A variant of Newton's method with accelerated third-order convergence. Applied Mathematics Letters 13 (8) (2000) 87-93. doi:http://dx.doi.org/10.1016/S0893-9659(00)00100-2. URL http://www.sciencedirect.com/science/article/pii/S0893965900001002

[2] F. J. Richard Burden, Numerical Analysis, 9th Edition, Brooks/Cole, Boston, MA, USA, 2004.

[3] Aungkulanon, P., Chai-Ead N., and Luangpaiboon P., "Simulated Manufacturing Process Improvement via Particle Swarm optimization and Firefly Algorithms". In Proc. International Multi Conference of Engineers and Computer Scientists IMECS'11, Hong Kong,2011,pp. 1123-1128.

[4] X. S. Yang, S. S. Hosseini, A. H. Gandomi, Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect, Applied Soft Computing, Vol. 12(3). 1180-1186(2012).

[5] A. Abdullah, S. Deris, M. S. Mohamad and S. Z. M. Hashim, A new hybrid firefly algorithm for complex and nonlinear problem, in: Distributed Computing and Artificial Intelligence. Advances in Intelligent and Soft Computing, 2012, Volume 151/2012, 673-680, doi:10.1007/978-3-642-28765-7_81

[6] T. C. Bora, L. S. Coelho, L. Lebensztajn, Bat-inspired optimization approach for the brushless DC wheel motor problems, IEEE Trans. Magnetics, Vol. 48 (2), 947-950 (2012).

[7] X.-S. Yang, Firefly algorithms for multimodal optimization, in: Proceedings of the 5th International Conference on Stochastic Algorithms: Foundations and Applications, SAGA'09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 169-178. URL http://dl.acm.org/citation.cfm?id=1814087.1814105

[8] Matej Črepinšek, Shih-Hsi Liu, and Marjan Mernik. 2013. Exploration and exploitation in evolutionary algorithms: A survey. ACM Comput. Surv. 45, 3, Article 35 (July 2013), 33 pages. DOI=10.1145/2480741.2480752 http://doi.acm.org/10.1145/2480741.2480752

[9] D. E. Goldberg, J. Richardson, Genetic algorithms with sharing for multimodal function optimization, in: Proceedings of the Second International Conference on Genetic Algorithms and Their Application, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1987, pp. 41-49. URL http://dl.acm.org/citation.cfm?id=42512.42519

[10] L. Duncan, The Weierstrass Function, University of Maine, 1901. URL https://books.google.lk/books?id=pEbDNwAACAAJ

[11] S. G. Krantz, Handbook of Complex Variables, Birkhuser, Boston, MA, USA, 1999.

[12] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[13] M. Dhivya and M. Sundarambal, Cuckoo search for data gathering in wireless sensor networks, Int. J. Mobile Communications, Vol. 9, pp. 642-656 (2011).