2016

# An Archived firefly Algorithm; A mathematical software to solve univariate nonlinear equations

M.K.A. Ariyaratne[1], T.G.I. Fernando[1] and S. Weerakoon[2]

Department of Computer Science, Faculty of Applied Sciences, University of Sri Jayewardenepura.[1]

Department of Mathematics, Faculty of Applied Sciences, University of Sri Jayewardenepura.[2]

**Abstract - The future of optimization is now being conquered by modern meta-heuristic algorithms. Genetic algorithms, differential evolution, harmony search, firefly algorithm and cuckoo search are such meta-heuristic algorithms which have marked their success over many optimization tasks. Simplicity of the algorithm, less memory consumption and the accuracy of the approximations can be stated as the major reasons for their popularity. In this article, we are presenting a software solution that proposes some modifications to the existing firefly algorithm. The modification; archived firefly algorithm [AFFA] exhibits the ability of finding almost all the real and complex roots of a given nonlinear equation within a reasonable range. The software implementation includes two main properties; an archive to collect the better fireflies and a flag to determine poor performance in firefly generations. The new modification is tested over Genetic algorithms (GA), a phenomenal in the field of nature inspired algorithms and also with a modified GA embedded with same properties the AFFA has. A simple GUI is developed using Matlab GUIDE to present the findings. Computer simulations show that the AFFA 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.**

**Index Terms - Firefly Algorithm, Nonlinear Equations, Archive, Real Roots, Complex Roots.**

## I. INTRODUCTION

Optimization leads the world for finding the best from being better. Hence the computer scientists are looking forward for finding various approaches that can contribute towards optimization. Natural optimization techniques are among them and are becoming popular due to the long lasting existence of the real world such natural practices. Genetic algorithms; which have come to the stage around 1970 going along with the theory of evolution by Charles Darwin, can be identified as one of the first such algorithms and still plays an important role among nature inspired algorithms [1, 2]. Ant colony optimization algorithms which mimic the ants' strange communication behavior are other popular algorithms which have been adopted for many real world optimization tasks [3, 4]. Bat inspired algorithm [5], Cuckoo birds algorithm [6], Firefly algorithm [7] are more recent nature inspired algorithms that represent different real world optimized phenomena. These algorithms can be classified in to two main fields as Evolutionary algorithms and Swarm intelligence algorithms. Evolutionary algorithms are population based algorithms which uses *mutation, recombination*, and natural selection to reproduce better generations [8]. Genetic algorithms, differential evolution [9] and genetic programming [10] are example evolutionary algorithms. Swarm intelligence, by its name mimics the collective behavior of different elements in the natural world. Particle swarm optimization [11], ant colony systems, firefly algorithms are some examples. One of the most advantageous properties of these algorithms is that most of them are the type of meta-heuristic. Therefore these algorithms can be adopted to solve a variety of optimization problems rather than heuristic algorithms. This paper aims to present the research carried out to find the accountability of using a modern nature inspired algorithm; firefly algorithm to solve univariate nonlinear equations having real and complex roots and the mathematical software that has been developed to accomplish the task.

## II. NONLINEAR EQUATIONS

Solving a nonlinear equation is finding $x$ which satisfies $f(x) = 0$, where $f(x)$ is nonlinear. Many numerical methods exist, but they have some major drawbacks like need of derivative information, strongly depends on the initial guess, inability of giving all the roots within an interval. When the equation is having either real and complex roots or complex roots only, the situation is even more difficult. Scientists have moved towards in finding better algorithms to minimize these drawbacks. But we have to ensure the meaningfulness of the computational effort as well. The algorithms should maintain the speed, accuracy and low memory consumption. The research literature reveals that there have been taken many steps to tune these existing approaches as well as finding new ways to solve nonlinear equations reducing the mentioned drawbacks.

A Variant of Newton's Method for Accelerated Third-Order Convergence is a suggested improvement by S. Weerakoon and T.G.I. Fernando [12]. The research involves changing the derivation of Newton's method. The original derivation involves approximating an indefinite integral of the derivative of the function by a rectangle. In their research they have modified it to be a trapezium so that the error of the approximation is reduced. They have shown that the order of convergence of the suggested modification is three and for some cases it is even higher than three. The main concern in this research was on accuracy of the approximation. But the mentioned drawbacks of numerical methods remain same.

Moving beyond numerical methods there can be seen a handful of research done with nature inspired algorithms to fulfill the task. Use of genetic algorithms to solve systems of nonlinear equations is addressed in some researches [13, 14, 15]. A hybrid algorithm implemented with Genetic algorithms and particle swarm optimization also has been tested in a research [16]. Applicability of harmony search in solving system of nonlinear equations is also addressed [17].

Heuristics were also in use of solving systems of nonlinear equations. One research has introduced the use of continuous global optimization heuristic called "continuous GRASP" to solve a nonlinear system [18]. They are addressing the problem of finding all the roots of a system of equations assuming that all roots are real.

Most of these approaches are focused on solving systems of nonlinear equations rather than a single equation. In almost all the research, they have deal only with the real roots. Finding all roots in a reasonable range within a single run is also left untouched.

Our problem of interest is to solve a univariate nonlinear equation having real and/or complex roots. We need to find all the roots within a reasonable interval/ reasonable range. Since we have touched the problem as an optimization problem, we can define the above problem 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 we treat the problem as an optimization problem, it becomes finding $x \in D$ $s.t. |f(x)| = 0$. However, since the function $f(x)$ may have multiple roots, the optimization problem $|f(x)| = 0$, also will have multiple optimal solutions. Our objective turns out to be finding all such optimal solutions.

## III. ARCHIVED FIREFLY ALGORITHM [AFFA]

AFFA is the software solution proposed in this paper to solve univariate nonlinear equations with real and/ or complex roots. Many nature inspired algorithms are good solution providers for various optimization tasks and they are very simple in the algorithms. That made researches easy to develop these algorithms for different optimization tasks. Firefly algorithm is a newly implemented such algorithm with the following assumptions about fireflies behavior [7].

1. An attraction of the fireflies to each other is gender independent.
2. Attractiveness is proportional to the brightness, for any two fireflies, the less brighter one will attracted and move towards the brighter one. this attraction decreases when distance increases. the brightest firefly will move randomly.
3. Brightness of a particular firefly is determined by its objective function.

The following pseudo code describes the original firefly algorithm and a simple 20 line Matlab program can implement the algorithm solving a mathematical optimization problem. Matlab as a mathematical software provides an easy environment to implement these types of algorithms.. It is also capable of creating user interfaces and enrich with plotting different functions and data so that the solutions can be graphically represented.

**Algorithm 1: Original Firefly Algorithm**

```
Begin;
Initialize algorithm parameters:
MaxGen: the maximum number of generations
$\gamma$: 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 $I_j > I_i$
          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.

$x_i = x_i + \beta(x_j - x_i) + \alpha(rand - 0.5); \rightarrow (1)$

Where

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

$\beta_0$ *is the attraction at* $r = 0$;

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

In the modified version we have added few more qualities to the original algorithm by introducing an archive and a flag. The new algorithm is named after its archiving property as Archived firefly algorithm [AFFA]. The pseudo code of the modified algorithm is as follows.

**Algorithm 2: Archived Firefly Algorithm**

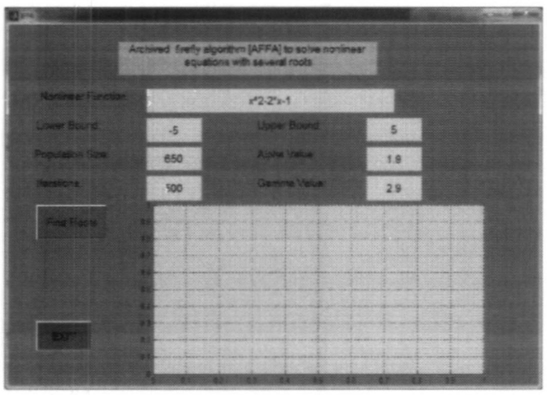
```
Begin;
Initialize algorithm parameters:
MaxGen: the maximum number of generations
$\gamma$: the light absorption coefficient
r: the particular distance between two fireflies
D: the domain space
Define the objective function $f(X)$,    where
        $X = (x_1, \ldots, x_d)^T$
Generate the initial 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
flag = true;
while flag = true and $t < MaxGen$ do
    for $i = 1 : n$ (all n fireflies) do
      for $j = 1 : n$ (all n fireflies) do
        if $I_j > I_i$
          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
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 archived firefly algorithm is specially designed to solve univariate nonlinear equations. Here finding a root of a nonlinear equation is considered as an optimization problem.

## IV. MATLAB SIMULATOR

A simulator is developed successfully to graphically represent the results of AFFA. To create GUIs, Matlab GUIDE (graphical user interface design environment) is used [21]. Matlab GUIDE provides tools for designing user interfaces for custom applications. Using the GUIDE Layout Editor, we have designed a simple user interface. GUIDE then automatically generated the MATLAB code for the user interface, and the user can modify the program to control the application.

The first step was taken to solve nonlinear equations with real roots only. The user has to provide the equation, the lower and the upper bounds to specify the interval to find roots and the values for problem specific parameters (s.t Population size, number of iterations, alpha and gamma values). Then the simulator will display the root approximations. A simple GUI in the Matlab environment is implemented to display results graphically.

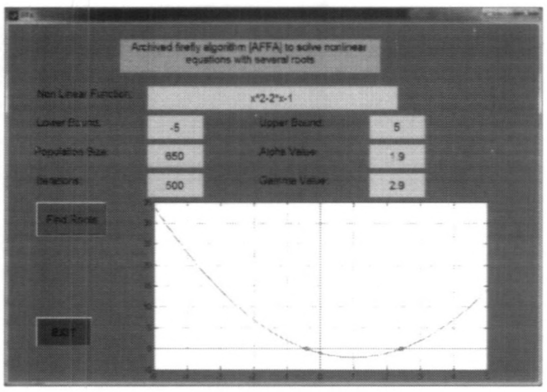

Archived firefly algorithm [AFFA] to solve nonlinear equations with several roots

Nonlinear Function: x^2-2*x-1

Lower Bound: -5 Upper Bound: 5

Population Size: 650 Alpha Value: 1.9

Iterations: 500 Gamma Value: 2.9


Archived firefly algorithm [AFFA] to solve nonlinear equations with several roots

Non Linear Function: x^2-2*x-1

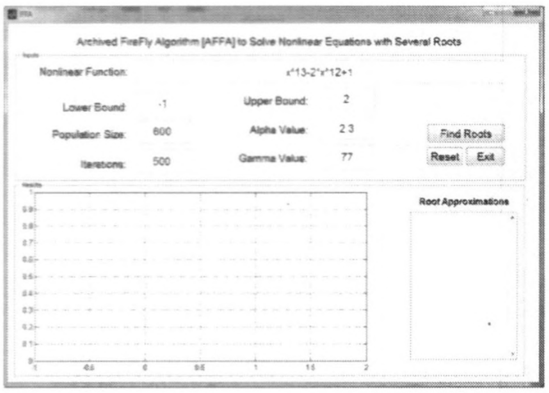
Lower Bound: -5 Upper Bound: 5

Population Size: 650 Alpha Value: 1.9

Iterations: 500 Gamma Value: 2.9

**Fig 1.** A Matlab GUI implemented to solve univariate nonlinear equations with real roots

When it comes to complex roots the environment is different. It is difficult to plot the function and display the roots; instead we used a real and an imaginary axis' to plot the roots. Apart from that we add a text box to display the root approximations. With all these modifications the new GUI to solve univariate nonlinear equations with real and/or complex roots is as follows.


Archived FireFly Algorithm [AFFA] to Solve Nonlinear Equations with Several Roots

Nonlinear Function: x^13-2*x^12+1

Lower Bound: -1 Upper Bound: 2

Population Size: 600 Alpha Value: 2.3

Iterations: 500 Gamma Value: 77

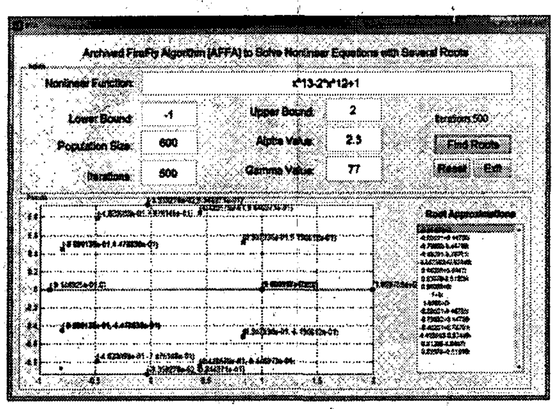
**Fig 2.** A Matlab GUI implemented to solve univariate nonlinear equations with real and/or complex roots

The new application displays the roots in an argand diagram and the root approximations are displayed in a list box. Because of the archiving property we can expect more than one approximation for an existing root. Apart from these, the application has a reset button to reset the inputs.

We have solved nonlinear equations having both real and complex roots using the new application.

$$y = x^{13} - 2x^{12} + 1 \rightarrow (3)$$

The above polynomial has 13 roots; 3 real roots and 10 complex roots. We have used our application to find the approximations with the accuracy of $10^{-3}$.



Archived Firefly Algorithm (AFFA) to Solve Nonlinear Equations with Several Roots

Nonlinear Function: x^13-2*x^12+1

Lower Bound: -1
Upper Bound: 2
Population Size: 600
Alpha Value: 2.5
Iterations: 500
Gamma Value: 77
Iterations 500

**Fig 3.** Root approximations given by the application for the Eqn 3.

## V. NUMERICAL EXAMPLES

Several nonlinear equations are tested against the proposed Matlab application of AFFA. Main concern is to identify different varieties of nonlinear equations where the modified algorithm can be applied. The following equations represent different types of nonlinear equations tested with the AFFA.

a. Following one dimensional trigonometric equation (adapted from Goldberg and Richardson, 1987) [19] has 51 real roots within the given interval and the AFFA approximated all 51 roots with an accuracy of $10^{-3}$. This equation proves the ability of AFFA in finding all the real roots of a nonlinear equation within a given interval.

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

b. Usual numerical methods need the equation to be differentiable to apply them for find roots. The Weierstrass functions possesses the property of being continuous everywhere but differentiable nowhere [20]. Therefore numerical approaches are not suitable for such situations. But the AFFA as a mathematical simulator for a modified nature inspired algorithm is capable of facing such situations successfully. Here we present an example Weierstrass function having 25 real roots within [-20, 20] interval.

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

c. Numerical methods also require the property of continuity of a nonlinear equation to be solved. $y = tan(x)$ is a popular nonlinear equation with discontinuities. Our solution does not need the continuity of the function and is capable of approximating all the roots in a large interval. (As an example we have used the interval [-40, 40].

$$y = \tan(x)$$

d. There are some nonlinear equations where there is additional difficulty in calculating the roots. These are functions in which the desired root has a ***multiplicity*** greater than 1. We can define it as follows. Let $\alpha$ be a root of the function $f(x)$, and imagine writing it in the factored form $f(x) = (x - \alpha)^m h(x)$, with some integer $m \geq 1$ and some continuous function $h(x)$ for which $h(\alpha) \neq 0$. Then we say that $\alpha$ is a root of $f(x)$ of multiplicity $m$. One of the main difficulties with the numerical calculation of multiple roots is

   a. Methods such as Newton's method and the secant method converge more slowly than for the case of a simple root.

We have solved the following nonlinear equation with multiple roots and found that our approach is successful.

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

e. The suggested application is capable of finding complex roots too. When the function is having both real and complex roots within a reasonable range, the application calculates accurate approximations. The following equations were tested using the software.

|  | 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**

As the interval for the real roots situations, for the complex roots we seek roots within the specified region. 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.

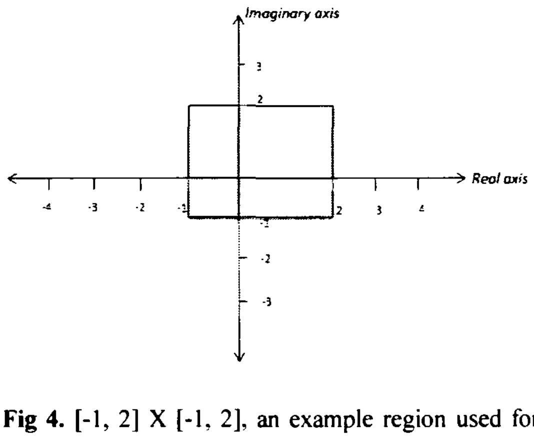


**Fig 4.** [-1, 2] X [-1, 2], an example region used for the root finding

## VI. RESULTS AND DISCUSSIONS

The quality of the developed software was compared with the original firefly algorithm, the original genetic algorithm and also with a genetic algorithm embedded with AFFA's qualities. For the comparative purpose we have kept the population size and the number of iterations per run 200, as a reasonable scale for real roots and for both real and complex situations we kept it to 600. We have run our application for 100 times (100 runs) and the average number of roots given by each algorithm, (maximum number of roots found / total roots)*100% are calculated. As an example a result 19, (49%) indicates 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 | AFFA |
|---|---|---|---|---|
| $y = \sin^3(5\pi x)$ | 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%) |

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

The real roots situations were smoothly handled by both Modified genetic algorithms and the proposed AFFA. The archiving property and diversifying the population during iterations is the reason for the good performance. When it comes to original algorithms, firefly algorithm performs better than GA.

| 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**

The modified genetic algorithm performed well for real roots but it fails in finding complex roots. 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. To check its availability over complex roots, we have to do more research on its recombination and selection parameters. With the obtained results we claim that AFFA under same conditions performed well for the complex roots as well. It is capable of handling nonlinear equations having both real and complex roots.

## VII. CONCLUSIONS

A new software application to find roots of a univariate nonlinear equation having real and complex roots is successfully implemented. Matlab, mathematical package is used to implement the application and graphical user interface design environment of matlab (GUIDE) is also applied successfully. This can be stated as the first mathematical software tool implemented to solve nonlinear equations using a nature inspired algorithm with a GUI component. The modification is successful in solving univariate nonlinear equations having both real and/or complex roots within a reasonable interval/ region. 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.

Although the implemented software application works well we would like to suggest some modifications to improve it further. The algorithm lacks a good parameter optimization. That is any user who uses the software has to enter algorithm specific parameters by themselves. Since the users are not experts in the field of nature inspired algorithms we can't expect them to enter good parameter values. In this initial step we have set all the parameters in to some reasonable values where most of the equations can deal with. But research is open to find methods to do a good parameter optimization for the algorithm for given situation.

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 is 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. The software with a graphical user interface gives users a user-friendly environment to use the application. But still this approach needs higher number of iterations and a large firefly population when we need higher accuracies in approximations. As an improvement we can test parameter optimization techniques which are able to adapt with our algorithm. Apart from that, one can check the application's ability of solving a given system of nonlinear equations.

## REFERANCES

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

[2] 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

[3] M. Dorigo et L.M. Gambardella, Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem, IEEE Transactions on Evolutionary Computation, volume 1, numéro 1, pages 53-66, 1997.

[4] M. Dorigo, Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italy, 1992.

[5] X. S. Yang, A New Metaheuristic Bat-Inspired Algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds. J. R. Gonzalez et al.), Studies in Computational Intelligence, Springer Berlin, 284, Springer, 65-74 (2010). http://arxiv.org/abs/1004.4170

[6] X.-S. Yang; S. Deb (December 2009). Cuckoo search via Lévy flights. World Congress on Nature & Biologically Inspired Computing (NaBIC 2009). IEEE Publications. pp. 210–214. arXiv:1003.1594v1

[7] Yang, X. S. (2009). "Firefly algorithms for multimodal optimization". Stochastic Algorithms: Foundations and Applications, SAGA 2009. Lecture Notes in Computer Sciences 5792. pp. 169–178. arXiv:1003.1466

[8] Bäck, T. (1996), Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms, Oxford Univ. Press.

[9] R. Storn, K. Price, Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization, 11 (4) (1997) 341-359. doi:10.1023/A:1008202821328. URL http://dx.doi.org/10.1023/A%3A1008202821328

[10] Poli, R.; Langdon, W. B. & McPhee, N. F. (2008), A field guide to genetic programming , Published via http://lulu.com and freely available at; http://www.gp-field-guide.org.uk .

[11] Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks IV. pp. 1942–1948. doi:10.1109/ICNN.1995.488968

[12] 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

[13] El-Emary, Ibrahiem MM, and MM Abd El-Kareem. "Towards using genetic algorithm for solving nonlinear equation systems." World Applied Sciences Journal 5.3 (2008): 282-289.

[14] Nikos E. Mastorakis. 2005. Solving non-linear equations via genetic algorithms. In Proceedings of the 6th WSEAS international conference on Evolutionary computing (EC'05), Ana Maria Madureira (Ed.). World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 24-28.

[15] Abolfazl Pourrajabian, Reza Ebrahimi, Masoud Mirzaei, Mehrzad Shams, Applying genetic algorithms for solving nonlinear algebraic equations, Applied Mathematics and Computation, Volume 219, Issue 24, 15 August 2013, Pages 11483-11494, ISSN 0096-3003, http://dx.doi.org/10.1016/j.amc.2013.05.057 (http://www.sciencedirect.com/science/article/pii/S0096300313006048)

[16] Arijit Biswas, Sambarta Dasgupta, Finding Solution Of Simultaneous Nonlinear Equations Using GA-PSO Hybrid Algorithm, Indian Conference on Trends in Modern Engineering Systems (ICONTIMES 2008), JIS College of Engineering, Kalyani, West Bengal, India.

[17] Ramadas, Gisela CV, and Edite Manuela da GP Fernandes. "Solving systems of nonlinear equations by harmony search." (2013).

[18] Michael J. Hirsch, Panos M. Pardalos, Mauricio G.C. Resende, Solving systems of nonlinear equations with continuous GRASP, Nonlinear Analysis: Real World Applications, Volume 10, Issue 4, August 2009, Pages 2000-2006, ISSN 1468-1218, http://dx.doi.org/10.1016/j.nonrwa.2008.03.006. (http://www.sciencedirect.com/science/article/pii/S1468121808000825)

[19] D. E. Goldberg, J. Richardson, Genetic algorithms with sharing for multimodal function optimization, in: Proceedings of the Second International Conference on Genetic Algorithms 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

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

[21] Matlab Help Files. GUI Building Basics", MathWorks. Url: http://in.mathworks.com/help/matlab/gui-building-basics.html