International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 2

A Study of Firefly Algorithm and its Application in Non-Linear Dynamic Systems

Gautam Mahapatra
Department of Computer Science and Engineering, Birla Institute of Technology, Jharkand, India

Srijita Mahapatra
Department of Electronics & Communication Engineering, UEM, Kolkata, India

Soumya Banerjee
Department of Computer Science and Engineering, Birla Institute of Technology, Jharkand, India

ABSTRACT

Firefly Algorithm (FA) is a newly proposed computation technique with inherent parallelism, capable for local as well as global search, meta-heuristic and robust in computing process. In this paper, Firefly Algorithm for Dynamic System (FADS) is a proposed system to find instantaneous behavior of the dynamic system within a single framework based on the idealized behavior of the flashing characteristics of fireflies. Dynamic system where flows of mass and / or energy is cause of dynamicity is generally represented as a set of differential equations and Fourth Order Runge-Kutta (RK4) method is one of used tool for numerical measurement of instantaneous behaviours of dynamic system. In FADS, experimental results are demonstrating the existence of more accurate and effective RK4 technique for the study of dynamic system.

Keywords: *FADS, Firefly Flashing, First Order Differential Equations, RK4*

Introduction

S. X. Yang of Cambridge University in the year 2008 has proposed new nature inspired, swarm intelligence based optimization method on the idealized behavior of the flashing characteristics of the common fireflies like Photinus [1].This algorithm assumes that each firefly is a solution of the optimization problem and quality of the solution is proportional to the intensity of glowing, and consequently brighter firefly attracts its partners which may explore the search space for the better solution. Yang has shown higher efficiency and effectiveness for different multi-dimensional, nonlinear and multimodal optimization problems [2]. In the year 2010 he has tested & designed several benchmark functions where this algorithm is outperforming other existing meta-heuristic algorithms [3]. Lukasik et al., 2009 has shown how FA performs better for continuous constraints optimization problems [4]. After the introduction this efficient computation technique it is found in literature that different applications[6] have developed to solve various common NP-Hard problems like data clustering[8], bi-level optimization[7], non-linear optimization problem [9], load forecasting problem [10] and multi-objective job-shop scheduling problem [12].Different improvements based chaos [11] and there is hybridization with support-vector machine (SVM) [10] have been also emerging as more efficient optimization technique. Yang has shown in Levy Flight distribution is outperforming than Gaussian distribution in randomized characteristics of fireflies [13].Dynamic systems are generally represented as a system of differential equation as parameters are varying with time and interrelated properties [14-20] and to measure different instantaneous property values the numerical solution of such differential equations are more practical and different numerical analysis are performed for the same [16,17]. In this present work FA has been studied and used to know the instantaneous behaviors of such dynamic systems modeled as system first order differential equations [18]. We are looking for numerical solutions, so we are focusing on numerical analysis based solutions and successfully proposed more accurate formula for the RK4which has been derived using Taylor Series[19] and embedded successfully in our proposed FADS.

Firefly Algorithm

Firefly Colony

Firefly Algorithm is most general optimization algorithm which can handle multi-dimensional, linear, non-linear, multimodal, non-convex, non-differentiable and continuous constraints optimization problem. Yang 2010 has shown that the Particle Swarm Optimization (PSO) is a special case it [3, 8]. FA has designed after inspiring on the unique short and rhythmic flashing pattern generated to attract mating partners and / or potential prey by the most of the fireflies' species. The radiation of this flashing light obeys the mathematical relation: $I \propto 1/_{r^2}$ where I is the intensity of flashing light at distance$r$, also due to absorption this intensity again decreases and combined effect limits the fireflies to communicate within few hundred meter of distance. Yang formulated this flashing behavior as an objective function and developed a new bio-inspired, swarm intelligence based meta-heuristic optimization algorithm.

Modeling the Attraction of Firefly

This flashing behavior of fireflies can be idealized and following three idealized rules for the firefly colony system were proposed by Yang 2010:

Rule I: All fireflies are unisex, so that one firefly is attracted to other fireflies regardless of their sex

Rule II: 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 moves randomly.

Rule III: The brightness or light intensity of a firefly is affected or determined by the landscape of the objective function to be optimized.

Using these three ideal rules and natural absorption of the media it is possible to formulate the attractiveness of the firefly as follows:

$$\beta(r) = \frac{\beta_0}{1+\gamma r^2} \approx \beta_0 e^{-\gamma r^2} \quad \text{(Gaussian form)} \quad (1)$$

where γ is the absorption co-efficient (or attenuation factor) of the media and β_0 is the initial attractiveness. From the inverse square law of light intensity it is possible to write for practical media:

$$I(r) = \frac{I_s}{1+\gamma r^2} \approx I_s e^{-\gamma r^2} \quad \text{(Gaussian form)} \quad (2)$$

Here one is added to the denominator to avoid the divisibility by zero and I_s is the intensity at light source.

Modeling the Movement of Firefly System

The Cartesian Distance between two fireflies F_i and F_j, are presently lying in n-dimensional search space at X_i and X_j respectively is given in following equation (3) and used in the new position calculation formula:

$$r_{ij} = \|X_i - X_j\| = \sqrt{\sum_{k=1}^{n}(x_{i,k} - x_{j,k})^2} \quad (3)$$

Fireflies are moving due to attractiveness generated for rhythmic pattern of flashing can be written as:

$$X_{i+1} = X_i + \beta_0 e^{-\gamma r_{i,j}^2}(X_j - X_i) + \alpha(rand - \tfrac{1}{2}) \quad (4)$$

Here firefly F_j is the brighter or more attractive than fireflyF_i, so firefly F_i is moving towards F_j and the position of F_i i.e. X_i is updating using equation (4) and it is also updating the intensity of F_i using equation (2) and hence the objective functions of the optimization problem. Attractiveness is updated using formula (1). For maximizing problems, intensity is directly proportional to the objective function value ($I(X) \propto f(X)$) and for minimization it will be inversely proportional ($I(X) \propto 1/f(X)$). In this position changing equation second part is due to attractiveness while the third part is for randomization, like mutation in GA [23] to avoid stuck-at-local optima problem, α is the randomization parameter and $rand$ is a random number generator uniformly distributed in$[0,1]$.

Algorithm for Simulation of Firefly Colony

Based on above mentioned modeling of attractiveness and then the movement of firefly due to this attractiveness with certain randomization due environmental interactions here is the pseudo code of the firefly system to solve the optimization problem.

Algorithm Firefly Algorithm (FA)
(* $max f(X)$ is the objective function, $X = (x_1, x_2, \ldots, x_n)$, position or solution vector in the constraint $n - dimensional$ search space*)

Step 1: [*Initialize*]
Create N number of Fireflies $F = (F_1, F_2, \ldots, F_N)$
Place them at random positions $S = (X_1, X_2, \ldots, X_N)$in the feasible solution region.
Evaluate intensities $I = (I_1, I_2, \ldots, I_N)$ for all N Fireflies
Initialize light absorption co-efficient (γ)
Set maximum number of generation (Gen_{Max})
Set current generation number $(t = 1)$

Step 2: [*Observe the swarming effect of the fireflies' colony for optimization*]
while$(t < Gen_{Max})$**do**
begin
    for$i = 1$**to**N**do**
    begin
        for$j = 1$**to**i**do**
        begin
        if$(I_j < I_i)$**then**
        begin
        2.1 Move F_j towards F_i in $n - dimensional$ space using (3)
        2.2 Update light intensity using and Evaluate new solutions (2)
        end
        Update attractiveness using (1)
        end
    end
    Rank the fireflies using Sort operation
    Find the current best firefly
    Move all fireflies randomly in the search space
    Update generation counter $t = t + 1$
end

Step 3: [*Finished*]
Return best Solution Vector and Optimized Value
Here the attractiveness depends on both flashing intensity and objective function value and distance among the fireflies which monotonically decreases with distance. But, for FA visibility is adjustable and more versatile in attractiveness variations, and this leads to higher mobility so the search space is explored more rapidly and execution time decreases.

Dynamic System

Dynamical system is a system in which any measuring parameter is varying with time. The world is full-of dynamic systems; all populations, human and otherwise, are dynamic systems, epidemics are dynamic systems, economies at all scales are dynamic systems, weather is complex dynamic system. Anywhere there are flows of mass and / or energy observed is a dynamic system. For the mathematical study of such varying system, instantaneous rate of changes parameters are modeled as system of time dependent Differential Equations. To measure any such variable property or characteristic we need to solve all such differential equations after imposing boundary conditions. In all cases solving such equations are not so simple and / or in some cases it is impossible and generally numerical approximate solutions are preferred over complex symbolic mathematical expression [16-20].Euler, Euler-Cauchy, Taylor Series, Runge-Kutta Second Order & Fourth Order etc. methods most common to use for numerical solution. Runge-Kutta Fourth Order Method is most popular method because it is advanced form of general Taylor Series Method [18-20] and mostly used for the dynamic systems which can be modeled as first order differential equations.

Example of Dynamic System

Fluid system in nature is very common dynamic system, in the year 1872, Joseph Boussinesq[21,22] has proposed very simplified model for water waves known as solitary wave or soliton still now it considered as reference and valid for weakly non-linear and fairly long waves. Following system of first order differential equations [equations no. (5) & (6)] known as the Boussinesq equations arerepresenting this model.

$$\frac{d\boldsymbol{D}}{dt} = \frac{g}{\rho_b}\widehat{\boldsymbol{e_3}} \times \boldsymbol{B} \quad (5)$$

$$\frac{d\boldsymbol{B}}{dt} = \frac{1}{2}\boldsymbol{D} \times \boldsymbol{B} \quad (6)$$

$\boldsymbol{D} = (D_1, D_2, D_3)^T$ - is horizontal displacement and$\boldsymbol{B} = (B_1, B_2, B_3)^T$ is vertical displacement, $\frac{g}{\rho_b}$is a non-dimensional constant and $\widehat{\boldsymbol{e_3}}=\begin{bmatrix}0 & -1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}$.

$$\frac{dD_1}{dt} = -\frac{g}{\rho_b}B_2 \quad (7)$$

$$\frac{dD_2}{dt} = \frac{g}{\rho_b}B_1 \quad (8)$$

$$\frac{dD_3}{dt} = 0 \quad (9)$$

$$\frac{dB_1}{dt} = \frac{1}{2}(D_2B_3 - D_3B_2) \quad (10)$$

$$\frac{dB_2}{dt} = \frac{1}{2}(D_3B_1 - D_1B_3) \quad (11)$$

$$\frac{dB_3}{dt} = \frac{1}{2}(D_1B_2 - D_2B_1) \quad (12)$$

In above differential equations[equation no. (7) – (12)] we are showing component of different velocities.For time dependent instantaneous behavior these equations must be solved and use of RK4 has been shown by Desale et al in the year 2013 [21].

Runge-Kutta Methodfor Dynamic Systems

To study the dynamic behavior of any system the gradient or variation relations of the system are expressed as a set of differential equation of equal or different orders. Most of the simple form is the first order form called as Ordinary Differential Equations (ODE). The solutions of these ODEs describe the functional behavior of the system and for this purpose different calculus oriented methods like integrating factor etc are used. The RK4 method is very efficient, as it avoids all kind of differentiations required in Taylor Series method. In the derivation of the RK4 method we get eleven independent algebraic relations for thirteen different coefficients and these are forming a system of equations with rank less than number unknowns. To find the value of unknowns Runge-Kutta had assigned empirically fixed values for two such variables and then solve the system of equation using numeric algorithm [19], finally we got the most popular recursive equation for evaluation.Let for a dynamic system there are N numbers of properties to measure and these are represented by the time dependent vector$Y(t) = \{y^i(t)\}_{i=1}^{N}$and $Y_0 = Y(t_0) = \{y^i(t_0)\}_{i=1}^{N}$ be initially measures or know parameters. Then the instantaneous dynamic behavior represented by 1st order differential equation form as follows:

$$\frac{dY(t)}{dt} = F\big(t, Y(t)\big) = \{f_i(t,\ Y(t))\}_{i=1}^{N} \text{ with initial condition } (t_0, Y_0) \quad (13)$$

And the solution will be of the form$Y = G(t) = \{g_i(t)\}_{i=1}^{N}$. Different complex forms of $F(t, Y(t))$ are possible and to get $G(t)$ different techniques are used and for numeric solution different approximate numeric algorithms are used. Most popular numeric method is RK4. This method is derived from Taylor Series method after approximating it up to 4th order term in the series and Runge-Kutta had assumed the recursive solution of the following form [19, 20]:

$$y_{k+1}^{i} = y_k^i + w_1 k_1^i + w_2 k_2^i + w_3 k_3^i + w_4 k_4^i, \text{ where} \quad (14)$$

$$k_1^i = hf_i(t_k, y_k^1(\mathrm{t}), y_k^2(\mathrm{t}), \dots, y_k^N(\mathrm{t})) \quad (15)$$

$$k_2^i = hf_i(t_k + a_1 h, y_k^1 + b_1 k_1^i, y_k^2 + b_1 k_1^i, \dots, y_k^N + b_1 k_1^i) \quad (16)$$

$$k_3^i = hf_i(t_k + a_2 h, y_k^1 + b_2 k_1^i + b_3 k_2^i, y_k^2 + b_2 k_1^i + b_3 k_2^i, \dots, y_k^N + b_1 k_2^i + b_3 k_2^i) \quad (17)$$

$$k_4^i = hf_i\big(t_k + a_3 h, y_k^1 + b_4 k_1^i + b_5 k_2^i + b_6 k_3^i, y_k^2 + b_4 k_1^i + b_5 k_2^i + b_6 k_3^i, \dots, y_k^N + b_4 k_1^i + b_5 k_2^i + b_6 k_3^i\big), \forall i = 1, \dots, N \quad (18)$$

After coefficient comparison with the Taylor's series method following eleven equations for thirteen unknowns$\{a_i\}_{i=1}^{3}, \{b_i\}_{i=1}^{6}$ and$\{w_i\}_{i=1}^{4}$of above equations (14) – (18) have generated, for which any conventional method is not suitable to solve.

$$b_1 = a_1, b_2 + b_3 = a_2, b_4 + b_5 + b_6 = a_3 \quad (19)\text{-}(21)$$

$$w_1 + w_2 + w_3 + w_4 = 1, w_2 a_1 + w_3 a_2 + w_4 a_3 = \tfrac{1}{2} \quad (22)\text{-}(23)$$

$$w_2 a_1^2 + w_3 a_2^2 + w_4 a_3^2 = \tfrac{1}{3}, w_2 a_1^3 + w_3 a_2^3 + w_4 a_3^3 = \tfrac{1}{4} \quad (23)\text{-}(25)$$

$$w_3 a_1 b_3 + w_4(a_1 b_5 + a_2 b_6) = \tfrac{1}{6}, w_3 a_1 a_2 b_3 + w_4 a_3(a_1 b_5 + a_2 b_6) = \tfrac{1}{8} \quad (26)\text{-}(27)$$

$$w_3 a_1^2 b_3 + w_4(a_1^2 b_5 + a_2^2 b_6) = \tfrac{1}{12}, w_4 a_1 b_3 b_6 = \tfrac{1}{24} \quad (28)\text{-}(29)$$

These eleven equations [equations no (19)-(29)]are forming a system of equations for which we have thirteen unknowns and eleven relations, so the rank is less than number of unknowns [18]. Runga-Kutta equalized the rank after assuming$a_1 = \frac{1}{2}, b_1 = 0$, then solved these equations and finally derived the popular recursive relation from equation (14) as:

$$y_{k+1}^{i} = y_k^i + \tfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (30)$$

with the solutions$a_2 = \frac{1}{2},\ a_3 = 1,\ b_2 = b_3 = \frac{1}{2}, b_4 = b_5 = 0, b_6 = 1, w_1 = w_4 = \frac{1}{6}, w_2 = w_3 = \frac{1}{3}$

Firefly Algorithm for Dynamic System (FADS)

For the design of FADS for general dynamic system we use these thirteen parameters as 13-dimensional position vector: $X = (a_1, a_2, a_3, b_1, b_2, \dots, b_6, w_1, w_2, w_3, w_4) \in R^{13}$for the different fireflies of FA. Above equations (19) to (29) are nothing but equality constraints to satisfyfor best possible position vector (X) to get the more accurate recursive formula represented in equation (14). For fitness or objective function calculation thesquare difference of left hand sidealgebraic expressions of these thirteen parameters with right hand side constant for these eleven equations for certain position vector or firefly is considered as square error δ_i and sum of these square errors as objective function and we have to minimize it towards zero value:

$$Minimize I(X) = \sum_{i=1}^{13} \delta_i^2 \qquad (31)$$

For **Algorithm** Firefly Algorithm (FA) mentioned above we have incorporated position vector of the fireflies along with square error measurement in equation (31) as intensity of these fireflies. After getting more accurate 13-parameters values recursive formula (14) with other formulas for the calculations of $k_i's$ mentioned equations (15)-(18) are used for numerical values of characteristics parameters on time for a dynamic system.

Experimental Results

Proposed FADS has been implemented using C++ programming language and tested with FA related parameters setting: Number of Firefly (N) = 20, Number of Generation (Gen_{Max}) = 1000, $\beta_0 = 0.2$, and $\gamma = 1.0$. This has been executed for 30 independent runs with said parameter values and we get different values for these thirteen variables which were found by Runge-Kutta and these are shown in Table 1. From this we can conclude that here is a new form of the recurrence relation which is outperforming the existing system.

Table 1.

Parameters	RK Solution	Average FADS Solution
a_1	0	**0.197491**
a_2	$\frac{1}{2}$	**0.665014**
a_3	1	**0.813665**
b_1	$\frac{1}{2}$	**0.201191**
b_2	$\frac{1}{2}$	**-0.737769**
b_3	$\frac{1}{2}$	**1.409750**
b_4	0	**1.005670**
b_5	0	**-0.864207**
b_6	1	**0.667915**
w_1	$\frac{1}{6}$	**0.109450**
w_2	$\frac{1}{3}$	**0.277182**
w_3	$\frac{1}{3}$	**0.362196**
w_4	$\frac{1}{6}$	**0.248217**

To study the performance of this dynamic system analyzer we have tested an expontial type system defined by the following first order differential equation:

$\frac{dY(t)}{dt} = \frac{dy(t)}{dt} = F\big(t, Y(t)\big) = ty(t)$ with initial condition (t_0, Y_0).

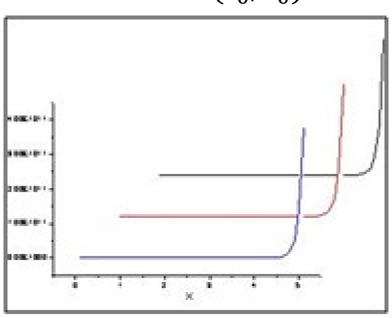

Fig. 1. 3D Comparative Study

The effectiveness of the system is presented by a comparative study of existing solution and standard C++ language library with the proposed FADS and this has been shown graphically in above Fig. 1.

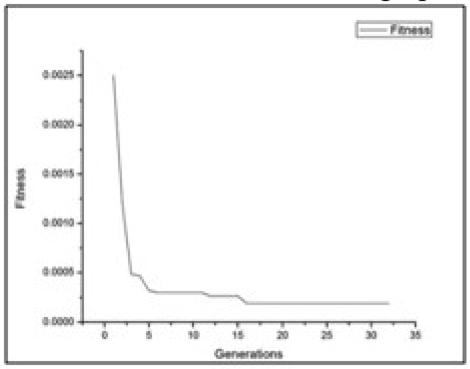

Fig 2: FA Convergence

The convergence of FA to achieve the best possible parameters values is also shown in Fig. 2.

Conclusion

This work has proposed how the robust and efficient FA optimization technique can be used to get numeric solutions of complex dynamic system represented by a set of differential equations form. This also suggests more accurate alternative of RK4 and also suggesting simpler way to develop RK-N where $N > 4$ for more accurate behavioral study of complex dynamic system.

Acknowledgements

The authors wish to acknowledge the support of the Post Graduate Teaching &Research Council of Asutosh College.

References

1. Yang, X. S., 2008. Nature-inspired meta-heuristic algorithms. Luniver press, UK.
2. Yang, X. S., 2009.Firefly algorithms for multimodal optimization. In *Stochastic algorithms: foundations and applications*, Springer Berlin Heidelberg, p. 169-178.
3. Yang, X. S., 2010. Firefly algorithm, stochastic test functions and design optimization. *International Journal of Bio-Inspired Computation*, *2*(2), 78-84.
4. Yang, X.S., 2010. Firefly algorithm. *Engineering optimization*, pp.221-230.
5. Łukasik, S., and Żak, S., 2009. Firefly algorithm for continuous constrained optimization tasks. In *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*. Springer Berlin Heidelberg, p. 97-106.
6. Yang, X.S. and He, X., 2013. Firefly algorithm: recent advances and applications. *International Journal of Swarm Intelligence*, *1*(1), pp.36-50.
7. Mahapatra, G. and Banerjee, S., 2014. Bilevel Optimization using Firefly Algorithm."*5th International Conference on Electronics Engineering & Computer Science (IEMCON)*. Elsevier Publications, pp. 313-322.
8. Senthilnath, J., Omkar, S.N. and Mani, V., 2011. Clustering using firefly algorithm: performance study. *Swarm and Evolutionary Computation*, *1*(3), pp.164-171.
9. Pal, S.K., Rai, C.S. and Singh, A.P., 2012. Comparative study of firefly algorithm and particle swarm optimization for noisy non-linear optimization problems. *International Journal of intelligent systems and applications*, *4*(10), p.50.
10. Kavousi-Fard, A., Samet, H. and Marzbani, F., 2014. A new hybrid modified firefly algorithm and support vector regression model for accurate short term load forecasting. *Expert systems with applications*, *41*(13), pp.6047-6056.
11. Gandomi, A.H., Yang, X.S., Talatahari, S. and Alavi, A.H., 2013. Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*, *18*(1), pp.89-98.
12. Karthikeyan, S., Asokan, P., Nickolas, S. and Page, T., 2015. A hybrid discrete firefly algorithm for solving multi-objective flexible job shop scheduling problems. *International Journal of Bio-Inspired Computation*, *7*(6), pp.386-401.
13. Yang, X.S., 2010. Firefly algorithm, Levy flights and global optimization. *Research and*

development in intelligent systems XXVI, pp.209-218.

14. Stuart, A.M., 1994. Numerical analysis of dynamical systems. *Acta numerica*, *3*, pp.467-572.
15. Joseph, D.D., 1966. Nonlinear stability of the Boussinesq equations by the method of energy. *Archive for Rational Mechanics and Analysis*, *22*(3), pp.163-184.
16. Guckenheimer, J., 2002. Numerical analysis of dynamical systems. *Handbook of dynamical systems*, *2*, pp.345-390.
17. Antoulas, A.C., 2005. *Approximation of large-scale dynamical systems*. Society for Industrial and Applied Mathematics.
18. S. Gilbert,2007, Linear Algebra and its Applications. Pacific Grove: Brooks Cole
19. Mathews, J.H., 2014. Numerical methods for mathematics, science and engineering. Prentice-Hall.
20. W. H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, 2007, Numerical Recipes The art of scientific computing, 3rd Edition, Cambridge University Press.
21. Desale, B.S. and Dasre, N.R., 2013. Numerical solution of the system of six coupled nonlinear ODEs by Runge-Kutta fourth order method. *Applied Mathematical Sciences*, *7*(6), pp.287-305.
22. Shi, F., Kirby, J.T., Harris, J.C., Geiman, J.D. and Grilli, S.T., 2012. A high-order adaptive time-stepping TVD solver for Boussinesq modeling of breaking waves and coastal inundation. *Ocean Modelling*, *43*, pp.36-51.
23. Goldberg, D.E., 2006. *Genetic algorithms*. Pearson Education India