I.J. Information Technology and Computer Science, **2014, 06, 19-24**
Published Online May 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2014.06.03

Chaotic Firefly Algorithm for Solving Definite Integral

Osama Abdel-Raouf
Department of Operations Research, Faculty of Computers and Information, Menoufia University, Menoufia, Shebin-El-come, Egypt
Email:osamaabd@hotmail.com

Mohamed Abdel-Baset
Department of Operations Research, Faculty of Computers and Informatics, Zagazig University, El-ZeraSquare, Zagazig, Sharqiyah, Egypt
Email:henawy2000@yahoo.com

Ibrahim El-henawy
Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, El-ZeraSquare, Zagazig, Sharqiyah, Egypt
Email:analyst_mohamed@yahoo.com

Abstract— In this paper, an Improved Firefly Algorithm with Chaos (IFCH) is presented for solving definite integral. The IFCH satisfies the question of parallel calculating numerical integration in engineering and those segmentation points are adaptive. Several numerical simulation results show that the algorithm offers an efficient way to calculate the numerical value of definite integrals, and has a high convergence rate, high accuracy and robustness.

Index Terms— Firefly Algorithm, Metaheurstic, Optimization, Chaos, Definite Integral

I. INTRODUCTION

The definite integral has wide ranging applications in operations research, computer science, mathematics, physical sciences and engineering. Numerical integration is the study of how the numerical value of an integral can be found. Which refers to find a square whose area is the same as the area under the curve, it is one of the classical topics of numerical analysis[1].The basic problem considered by numerical integration is to compute an approximated solution to a definite integral $\int_a^b f(x)dx$. Situations arise which the analytical method developed so far cannot be used to evaluate some definite integrals. For example, an integrand may not have an obvious anti-derivative such as cos^2x and $1/(lin x)$ or maybe the integrand represented by individual data points, which makes finding an anti-derivative impossible. When analytical methods fail, we often turn to numerical methods [2], which are typically done on calculator or computer.

These methods do not produce exact values of definite integrals, but provide approximations that are generally accurate. Briefly, some of the more advanced methods for which software is widely available are:

1.1 Midpoint rule

Suppose f is defined and integrable on [a, b]. The midpoint rule approximation to $\int_a^b f(x)dx$ using n equally spaced subintervals on [a,b] is [3]:

$$M(n) = f(m_1)\Delta x + f(m_2)\Delta x + \cdots + f(m_n)\Delta x \quad (1)$$

Where $\Delta x= (b-a)/n$, $x_k=a+k\Delta x$, *and* m_k is midpoint of $[x_{k-1},x_k]$, *for* $k=1,2,\dots,n$.

1.2 Trapezoid rule

Another method for estimating $\int_a^b f(x)dx$ is trapezoid rule [3], suppose f is defined and integrable on [a, b]. The trapezoid rule approximation to $\int_a^b f(x)dx$ using n equally spaced subintervals on [a,b] is :

$$T(n) = (\frac{1}{2}f(x_0) + \sum_{k=1}^{n-1} f(x_k) + \frac{1}{2}f(x_n))\Delta x \qquad (2)$$

where $\Delta x=(b-a)/n$ and $x_k=a+k\Delta x$, for $k=0,1,\dots,n$.

1.3 Simpson's rule

Suppose f is defined and integrable on [a,b]. The Simpson's Rule approximation to $\int_a^b f(x)dx$ using n equally spaced subintervals on [a,b] is [3]:

$$S(n) = [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 4f(x_{n-1}) + f(x_n)]\frac{\Delta x}{3} \qquad (3)$$

Where n is an even integer, $\Delta x=(b-a)/n$, and $x_k=a+k\Delta x$, for k= 0,1,2,…,n.

1.4 Newton –Cotes formula

The Newton-Cotes formulas are the most common numerical integration methods [1-4]. They are based on the strategy of replacing a complicated function with an approximating function that is easy to integrate.

$$I = \int_a^b f(x)d(x) \cong \int_a^b f_n(x)d(x) \qquad (4)$$

Where

$$f_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1}x^{n-1} + a_n x^n \qquad (5)$$

Let $x_0, x_1, \cdots, x_n$ be given distinct nodes in closed interval [a, b]. We want to determine constants $A_0, A_1, \cdots, A_n$ such that

$$\int_a^b f(x) = A_0 f(x_0) + \cdots + A_n f(x_n) \qquad (6)$$

For any polynomial f of degree $\leq n$. Strictly speaking, In Newton-Cotes Integration we used points that were equally spaced. However, there was no need for the points to have any special spacing.

1.5 Clenshaw–Curtis integration

Newton–Cotes formulas with equally spaced abscissas are of practical use only for small point numbers, say $n \leq 8$. For n as low as nine, the coefficients *Ai* have different signs. As n increases, the coefficients A_i become large in absolute value, leading to unstable evaluation of the integral [1-4]. This problem can be avoided by choosing the abscissas in a more sophisticated way.

If n is even, then the Clenshaw–Curtis formula can be written

$$\int_a^b f(x)dx \approx \frac{b-a}{2}\left[a_0 - \frac{2a_2}{(1)(3)} - \frac{2a_4}{(3)(5)} - \ldots - \frac{2a_{n-2}}{(n-3)(n-1)} - \frac{a_n}{(n-1)(n+1)}\right] \qquad (7)$$

Like other formulas of the Newton–Cotes type, Clenshaw–Curtis will integrate exactly polynomials of order n or less. In practice, it does rather better than other rules of the same order, because of the bounded variation properties of Chebyshev polynomials.

1.6 Gaussian quadrature

In Newton-Cotes Integration we used points that were equally spaced. However, there was no need for the points to have any special spacing [1-4]. If we wish to estimate the integral $\int_a^b f(x)$ And if we have any set of points $\{x_0, x_1, \ldots, x_n\}$ and $n+1$ coefficients A_i, then we can estimate the integral by the formula.

$$\int_a^b f(x)g(x)dx \approx A_0 f(x_0) + \cdots + A_n f(x_n) \qquad (8)$$

where $g(x)$ is a weight function which is greater than zero on the interval $[a, b]$. The correct choice for $x_0, \ldots, x_n$ turns out to be the zeros of an orthogonal polynomial $P\{n+1\}$ of order $n + 1$. An important point is that the coefficients A_i are positive. Moreover, $A_0 + A_1 + \cdots + A_n = \int_a^b g(x)dx$ so no coefficient can be larger than the summation of A_i .

1.7 Monte Carlo method

It means using random numbers in scientific computing. More precisely, it means using random numbers as a tool to compute something that is not random. The idea of estimating an integral by random sampling is a natural one in a statistical context [5].

In Monte Carlo method, points $x_1, \ldots, x_n$ are chosen randomly in the integration region and the integral is estimated by

$$\bar{f} = \frac{V}{n}\sum_{i=1}^{n} f(x_i) \qquad (9)$$

where V is the volume of the integration region. Convergence is certain almost definitely by the central limit theorem under very weak conditions on *f*.

However, these traditional methods have limitations: the rectangle rule method, trapezoidal rule method, Simpson's rule method are suitable for the bad smooth integrand, but their precisions are low; The Newton-Cotes method is one of the constructing integrand based on the interpolating functions, but the convergence is not guaranteed for higher order; The Newton-Cotes method, for the Romberg method and Gauss method, their convergent speeds are quick and the computational precisions are high, but their computations are complex[1-5].

This paper is organized as follows: after introduction, the original firefly algorithm is briefly introduced in section 2. Section 3 introduces the meaning of chaos. In section 4, the proposed algorithm is described, while the results are discussed in section 5. Finally, conclusions are presented in section 6.

II. Firefly Algorithm

Firefly algorithm is one of the latest additions to the family of swarm intelligence metaheuristics for optimization problems. It was proposed by Yang in 2009 [6] and it has since then been applied in several applications because of its few parameters to adjust, easy to understand, realize, and compute, it was applied to various fields, such as codebook of vector quantization [7], in-line spring-mass systems [8]; mixed variable structural optimization [9]; nonlinear grayscale image enhancement [10], travelling salesman problems [11], continuously cast steel slabs [12], promoting products online [13], nonconvex economic dispatch problems [14], chiller loading for energy conservation [15], stock market price forecasting [16], and multiple objectives optimization [17]. Although the algorithm has many similarities with other swarm based algorithms such as Particle Swarm Optimization [18], Artificial Bee Colony Optimization [19]and Ant Colony Optimization [6], the FA has proved to be much simpler both in concept and implementation and has better performance compared to the other techniques.

2.1 Flashing behaviour of fireflies

The FA was based on the flashing patterns and behaviour patterns of the fireflies. The fireflies use the flashing patterns to communicate with each other. Yang did not mimic their behaviour in full detail, but created a simplified algorithm based on the following three rules:

1. All fireflies are unisexual, so that one firefly will be attracted to other fireflies regardless of their sex;
2. Attractiveness is proportional to the firefly's brightness; for any couple of flashing fireflies, the less bright one will move towards the brighter one; attractiveness is proportional to the brightness which decreases with increasing distance between fireflies; if there are no brighter fireflies than a particular firefly, this individual will move randomly in the space;
3. The brightness of a firefly is somehow related to the analytical form of a cost function; for a maximization problem, brightness can be proportional to the value of the cost function; other forms of brightness can be defined in a similar matter to the fitness function in genetic algorithms.

2.2 Attractiveness and light intensity

In the algorithm, two important factors are involved: the variation of light intensity and the formulation of the attractiveness. For example, suppose that the attractiveness of a firefly is determined by its brightness, which in turn is associated with the encoded objective function,then the higher of the brightness and, the better the location and the more fireflies will be attracted to the direction. However, if the brightness is equal, the fireflies will move randomly. As light intensity and thus attractiveness decreases as the distance from the source increases, the variations of light intensity and attractiveness should be monotonically decreasing functions.

In order to implement FA, there are some definitions:

Definition 1: the variation of light intensity;

We know, the light intensity varies according to the inverse square law

$$I(r) = {}^{I_s}/_{r^2} \tag{10}$$

Where I(r) is the light intensity at a distance r and Is is the intensity at the source.

When the medium is given, the light intensity can be determined as follows:

$$I(r) = I_0 e^{-\gamma r} \tag{11}$$

To avoid the singularity at r=0 in (1), the equations can be approximated in the following Gaussian form:

$$I(r) = I_0 e^{-\gamma r^2} \tag{12}$$

Where γ is light absorption coefficient.

Definition 2: formulation of the attractiveness

As firefly attractiveness is proportional to the light intensity seen by adjacent fireflies, we can now define the attractiveness β of a firefly by

$$\beta = \beta_0 e^{-\gamma r^2} \tag{13}$$

Where β_0 is the attractiveness at r=0.

Definition 3: formulation of location moving

$$x_i(t+1) = x_i(t) + \beta\left(x_j(t) - x_i(t)\right) + \alpha\varepsilon_i \tag{14}$$

Where $x_i(t+1)$ is the position of x_i after *t+1* times movements; α is the step parameter which varies between [0,1] ; ε_i is a random factor conforming Gaussian distribution between[0,1].

The basic steps of the FA are summarized as the pseudo code shown in Fig. 1 which consists of the three rules discussed above.

firefly algorithm
Begin
Objective function $f(x)$, $x = (x_1, \dots, x_d)^T$
Generate initial population of n fireflies x_i, $i = 1, 2, \dots, n$
Formulate light intensity I so that it is associated with $f(x)$
While *(t<MaxGeneration)*
Define absorption coefficient γ
for *$i = 1 : n$($n$fireflies)*
for *$j = 1 : n$($n$fireflies)*
if *($I_j > I_i$),*
move firefly i towards j
end if
Vary attractiveness with distance r via $e^{-\gamma r^2}$
Evaluate new solutions and update light intensity
end for *j*
end for *i*
Rank the fireflies and find the current best
end while
Post-processing the results and visualization
End

Fig. 1. Pseudo code of the firefly algorithm

III. CHAOS

Generating random sequences with a long period, and a good consistency is very important for easily simulating complex phenomena, sampling, numerical analysis, decision making and especially in heuristic optimization [20]. Its quality determines the reduction of storage and computation time to achieve the desired accuracy [21]. Chaos is a deterministic, random-like process found in nonlinear, dynamical system, which is non-period, non-converging and bounded. Moreover, it depends on its initial condition and parameters [22-24]. Applications of chaos in several disciplines including operations research, physics, engineering, economics, biology, philosophy and computer science[25-27].

Recently chaos is extended to various optimization areas because it can more easily escape from local minima and improve global convergence in comparison with other stochastic optimization algorithms [28-34]. Using chaotic sequences in Firefly Algorithm can be helpful to improve the reliability of the global optimality, and also enhance the quality of the results.

3.1 Chaotic maps

At random-based optimization algorithms, the methods using chaotic variables instead of random variables are called chaotic optimization algorithms (COA) [34]. In these algorithms, due to the non-repetition and ergodicity of chaos, it can carry out overall searches at higher speeds than stochastic searches that depend on probabilities [43-48]. To resolve this issue, herein one-dimensional and non-invertible maps are utilized to generate chaotic sets. We will illustrate some of well-known one-dimensional maps as:

3.1.1 Logistic map

The Logistic map is defined by:

$$Y_{n+1} = \mu Y_n(1 - Y_n)\ Y \in (0,1)\ 0 < \mu \leq 4 \tag{15}$$

3.1.2 Sine map

The Sine map is written as the following equation:

$$Y_{n+1} = \frac{\mu}{4}\sin(\pi Y_n)\ Y \epsilon\ (0,1)\ 0 < \mu \leq 4 \tag{16}$$

3.1.3 Iterative chaotic map

The iterative chaotic map with infinite collapses is described as:

$$Y_{n+1} = \sin\left(\frac{\mu\pi}{Y_n}\right)\ \mu \in (0,1) \tag{17}$$

3.1.4 Circle map

The Circle map is expressed as:

$$Y_{n+1} = Y_n + \alpha - \left(\frac{\beta}{2\pi}\right)\sin(2\pi Y_n)\ mod\ 1 \tag{18}$$

3.1.5 Chebyshev map

The family of Chebyshev map is written as the following equation:

$$Y_{n+1} = \cos(k cos^{-1}(Y_n))\ \ Y \in (-1,1) \tag{19}$$

3.1.6 Sinusoidal map

This map can be represented by

$$Y_{n+1} = \mu Y_k^2 \sin\ (\pi Y_n) \tag{20}$$

3.1.7 Gauss map

The Gauss map is represented by:

$$Y_{n+1} = \begin{cases} 0 & Y_n = 0 \\ \frac{\mu}{Y_n}\ mod\ 1 & Y_n \neq 0 \end{cases} \tag{21}$$

3.1.8 Sinus map

Sinus map is formulated as follows:

$$Y_{n+1} = 2.3(Y_n)^{2\sin(\pi Y_n)} \tag{22}$$

3.1.9 Dyadic map

Also known as the dyadic map bit shift map, 2*x* mod 1 map, Bernoulli map, doubling map or saw tooth map. Dyadic map can be formulated by a mod function:

$$Y_{n+1} = 2Y_n\ mod\ 1 \tag{23}$$

3.1.10 Singer map

Singer map can be written as:

$$Y_{n+1} = \mu(7.86Y_n - 23.31Y_n^2 + 28.75Y_n^3 - 13.3Y_n^4) \tag{24}$$

μ between 0.9 and 1.08

3.1.11 Tent map

This map can be defined by the following equation:

$$Y_{n+1} = \begin{cases} \mu Y_n & Y_n < 0.5 \\ \mu(1 - Y_n) & Y_n \geq 0.5 \end{cases} \tag{25}$$

IV. The Proposed Algorithm (IFCH) For Solving Definite Integral

Suppose that segmentation S splits an integral interval [a,b] into n-subintervals:

$[x_0,x_1],[x_1,x_2],\ldots,[x_{k-1},x_k],[x_{n-1}, x_n]$, where $x_j< x_{j+1}$ *for j = 1,2,…,n-1; $x_0 = a$, and $x_n= b$*, also define $\Delta x_k= x_k - x_{k-1}$ *for k = 1,2,…,n.* Using this notation, the integral *f(x)* in [a,b] can be approximated as[36]:

$$\int_a^b f(x) \approx \sum_{k=1}^{n} \frac{1}{6}\begin{pmatrix} f(x_{k-1}) \\ +4f\left(\frac{x_{k-1}+x_k}{2}\right) \\ +f(x_k) \end{pmatrix}\Delta x_k \tag{26}$$

In the proposed chaotic Firefly Algorithm, we used chaotic maps to tune the Firefly Algorithm parameters and improve the performance [20]. The steps of the proposed chaotic firefly algorithm for solving definite integral are as follows:

Step 1 Generate the initial population of fireflies, $\{x_1, x_2, x_3, \ldots, x_n\}$

Step 2 Compute intensity for each firefly member,$\{I_1, I_2, I_3, \ldots, I_n\}$

Step 3 Calculate the parameters (β , γ) using the following Sinusoidal map[35]:

$$Y_{n+1} = \cos(k\cos^{-1}(Y_n))\ Y \in (-1,1)$$

where n is the iteration number.

Step 4 Move each firefly xi towards other brighter fireflies. The position of each firefly is updated by

$$x_i(t+1) = x_i(t) + \beta_0 e^{-\gamma r^2}\left(x_j(t) - x_i(t)\right) + \alpha\varepsilon_i$$

Where α computed by the following randomness equation as shown below:

$$\alpha^i = \alpha_{max} - (\alpha_{max} - \alpha_{min})\left(\frac{I^i_{max} - I^i_{mean}}{I^i_{max} - I^i_{min}}\right) \tag{6}$$

In this equation α^i represents randomness parameters at cycle *i*. α_{max} and α_{min} represent maximum and minimum randomness parameters defined in the algorithm respectively. I^i_{max} and I^i_{min} represent maximum light intensity, minimum light intensity and mean value of light intensity of all fireflies at cycle i respectively.

Step 5 Update the solution set.

Step 6 Terminate if a termination criterion is fulfilled; otherwise go to Step 2

V. Numerical Results

Several examples have been given to verify the weight of the proposed algorithm. The initial parameters are set at n= 40; maximum iteration number = 100;$\alpha_{\max} = 0.8$; $\alpha_{\min} = 0.1$.

The results of IFCH algorithm are conducted from 30 independent runs for each integrand. The selected chaotic map for all examples is the Sinusoidal map for β , γ values, and randomized for α values, whose equations is shown above.

All the experiments were performed on a Windows 7 Ultimate 64-bit operating system; processor Intel Core i5 760 running at 2.81 GHz; 4 GB of RAM and code was implemented in C#.

The integral values of functions x^2e^{-x},cos^2x , sin^2x in [0,2] ; $1/1+x^2$ in [0,1] ; $1/x$ in [1,2] and $1/linx$ in [2,3] are selected for experiments.

Table1. The best results of selected functions

F(x)	Monte Carlo Method	Midpoint Rule	Trapezoidal Rule	Simpson's Rule	IFCH	Exact Value
x^2e^{-x}	0.646649	0.646659	0.646633	0.646651	**0.646647**	**0.646647**
sin^2x	1.1892	1.19047	1.18667	1.1890	**1.1892**	**1.1892**
$\frac{1}{1+x^2}$	0.785396	0.786231	0.783732	0.785395	**0.785398**	**0.785398**
Cos^2x	1.9766	2.5017	1.9989	1.9990	**1.0174**	**1.0174**
$\frac{1}{x}$	0.6924	0.6912	0.6970	0.6933	**0.6931**	**0.6931**
$\frac{1}{linx}$	1.1187	1.4164	1.1223	1.1185	**1.1184**	**1.1184**

The results of IFCH algorithm are privileged compared with the results of the Monte Carlo method, trapezoidal rule, Simpson's rule and midpoint rule. In comparison with exact values, we find that the results of IFCH algorithm are very close to the exact values of the selected functions under study. If a large number of well-behaved one-dimensional integrands are to be integrated, and the user is willing to do some analytic analysis to obtain efficiency, then it is hard to go past the classical methods. Usually though, users will choose to use IFCH algorithm, to save time and to gain reliability.

The reason for getting better results than the other algorithms considered is that the search power of Firefly Algorithm. Adding to this, using chaos helps the algorithms to escape from local solutions.

VI. Conclusions

This paper introduced an improved Firefly Algorithm by blending with chaos for calculation the numerical value of definite integrals. This algorithm has the ability to trounce the shortage that the segmentation points are uniform in traditional methods. Many simulation examples show that the algorithm can converge to the best solution, and it has a high convergence rate and high accuracy.

Acknowledgments

The authors would like to thank the anonymous reviewers for their careful reading of this paper and for their helpful comments.

References

[1] S. S. Rao, *Applied numerical methods for engineers and scientists*: Prentice Hall Professional Technical Reference, 2001.

[2] E. Kreyszig, *Advanced engineering mathematics*: Wiley. com, 2007

[3] P. J. Davis and P. Rabinowitz, *Methods of numerical integration*: Courier Dover Publications, 2007.

[4] G. E. Forsythe, M. A. Malcolm, and C. B. Moler, *Computer methods for mathematical computations*: Prentice Hall Professional Technical Reference, 1977.

[5] J. Hammersley and D. Handscomb, *Monte Carlo Methods*. Methuen, London, 1964 ..

[6] X.-S. Yang, *Nature-inspired metaheuristic algorithms*: Luniver Press, 2010.

[7] M.-H. Horng and T.-W. Jiang, "The codebook design of image vector quantization based on the firefly algorithm," in *Computational Collective Intelligence. Technologies and Applications*, ed: Springer, 2010, pp. 438-447.

[8] R. Dutta, R. Ganguli, and V. Mani, "Exploring isospectral spring–mass systems with firefly algorithm," *Proceedings*

of the Royal Society A: Mathematical, Physical and Engineering Science, vol. 467, pp. 3222-3240, 2011.
[9] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Mixed variable structural optimization using firefly algorithm," *Computers & Structures,* vol. 89, pp. 2325-2336, 2011.
[10] T. Hassanzadeh, H. Vojodi, and F. Mahmoudi, "Non-linear grayscale image enhancement based on firefly algorithm," in *Swarm, Evolutionary, and Memetic Computing*, ed: Springer, 2011, pp. 174-181.
[11] G. K. Jati, "Evolutionary discrete firefly algorithm for travelling salesman problem," in *Adaptive and Intelligent Systems*, ed: Springer, 2011, pp. 393-403.
[12] O. K. K. L. JEKLENE, "Optimization of the Quality of Continuously Cast Steel Slabs Using the Firefly Algorithm," *Materiali in tehnologije,* vol. 45, pp. 347-350, 2011.
[13] H. Banati and M. Bajaj, "Promoting products online using firefly algorithm," in *Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on*, 2012, pp. 580-585.
[14] X.-S. Yang, S. S. Sadat Hosseini, and A. H. Gandomi, "Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect," *Applied Soft Computing,* vol. 12, pp. 1180-1186, 2012.
[15] L. d. S. Coelho and V. C. Mariani, "Improved firefly algorithm approach for optimal chiller loading for energy conservation," *Energy and Buildings,* 2012.
[16] A. Kazem, E. Sharifi, F. K. Hussain, M. Saberi, and O. K. Hussain, "Support vector regression with chaos-based firefly algorithm for stock market price forecasting," *Applied Soft Computing,* 2012.
[17] X.-S. Yang, "Multiobjective firefly algorithm for continuous optimization," *Engineering with Computers,* pp. 1-10, 2013.
[18] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm intelligence,* vol. 1, pp. 33-57, 2007.
[19] P.-W. TSai, J.-S. Pan, B.-Y. Liao, and S.-C. Chu, "Enhanced artificial bee colony optimization," *International Journal of Innovative Computing, Information and Control,* vol. 5, pp. 5081-5092, 2009.
[20] B. Alatas, "Chaotic harmony search algorithms," *Applied Mathematics and Computation,* vol. 216, pp. 2687-2699, 2010.
[21] W. Gong and S. Wang, "Chaos Ant Colony Optimization and Application," in *Internet Computing for Science and Engineering (ICICSE), 2009 Fourth International Conference on*, 2009, pp. 301-303.
[22] B. Alatas, "Chaotic bee colony algorithms for global numerical optimization," *Expert Systems with Applications,* vol. 37, pp. 5682-5687, 2010.
[23] A. Gandomi, X.-S. Yang, S. Talatahari, and A. Alavi, "Firefly algorithm with chaos," *Communications in Nonlinear Science and Numerical Simulation,* vol. 18, pp. 89-98, 2013.
[24] J. Mingjun and T. Huanwen, "Application of chaos in simulated annealing," *Chaos, Solitons & Fractals,* vol. 21, pp. 933-941, 2004.
[25] L. d. S. Coelho and V. C. Mariani, "Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization," *Expert Systems with Applications,* vol. 34, pp. 1905-1913, 2008.
[26] M. S. Tavazoei and M. Haeri, "Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms," *Applied Mathematics and Computation,* vol. 187, pp. 1076-1085, 2007.
[27] R. Hilborn, *Chaos and nonlinear dynamics: an introduction for scientists and engineers*: oxford university press, 2000.
[28] D. He, C. He, L.-G. Jiang, H.-W. Zhu, and G.-R. Hu, "Chaotic characteristics of a one-dimensional iterative map with infinite collapses," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on,* vol. 48, pp. 900-906, 2001.
[29] A. Erramilli, R. Singh, and P. Pruthi, *Modeling packet traffic with chaotic maps*: Citeseer, 1994.
[30] R. M. May, "Simple mathematical models with very complicated dynamics," in *The Theory of Chaotic Attractors*, ed: Springer, 2004, pp. 85-93.
[31] A. Wolf, "Quantifying chaos with Lyapunov exponents," *Chaos,* pp. 273-290, 1986.
[32] R. L. Devaney, "An introduction to chaotic dynamical systems," 2003.
[33] C. Letellier, *Chaos in nature* vol. 81: World Scientific Publishing Company, 2013.
[34] R. Barton, "Chaos and fractals," *The Mathematics Teacher,* vol. 83, pp. 524-529, 1990.
[35] I. Fister, I. Jr Fister, X.-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms," *Swarm and Evolutionary Computation,* 2013.
[36] L. Qu and D. He, "Solving Numerical Integration by Particle Swarm Optimization," in *Information Computing and Applications*, ed: Springer, 2010, pp. 228-235.

Authors' Profils

Ibrahim El-henawy received the M.S. and Ph.D. degrees in computer science from State University of New York, USA in 1980 and 1983, respectively. Currently, he is a professor in computer science and mathematics department, Zagazig University. ibrahim is a member of IEEE and AAEE (Australasian Association for Engineering Education) ,He is also a reviewer in different international journals and conferences. He has published more than 200 refereed articles in National and international journals, edited books, and conference proceedings. His current research interests are mathematics, operations research, statistics, networks, optimization, Intelligent Computing, Computer Theory, digital image processing, and security

Osama Abdel-Raouf received the M.S. and Ph.D. degrees in operations research and decision support systems from Monofia University. Currently, he is an associate professor in operations research department, Monofia University. His current research interests are evolutionary algorithms, artificial intelligence, and decision support systems.

Mohamed Abd El-Baset received the B.Sc. degree in information system and technology from Zagazig University in 2006, and he obtained his M.S. degree in operations research and decision support systems from Zagazig University, in 2011. Currently, he is a teaching assistant in operations research department, Zagazig University. His current research interests are characterization of probability distribution, optimization, Intelligent Computing, Evolutionary Computation and decision support systems.