



A CHAOTIC BAT ALGORITHM FOR SOLVING DEFINITE INTEGRAL

Khalil AL-Wagih*

Faculty of Computer science & Information System,

Thamar University, Thamar, Republic of Yemen

E-mail: khalilwagih@gmail.com

ABSTRACT

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

Indexing terms/Keywords

Bat algorithm; meta-heuristics; optimization; chaos; definite integral.

SUBJECT CLASSIFICATION

Computer science & Information System



Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol .14, No.4

www.ijctonline.com , editorijctonline@gmail.com



INTRODUCTION

The definite integral has wide ranging applications in operation researching, computer science, mathematics, physical sciences and engineering. Numerical integration is the study of how numeric values of an integral can be found. Which refers to finding a square whose area is the same as the area under a curve, it is one of the classical topics of numerical analysis [1]. The basic problem considered by numerical integration is to compute an approximate solution to a definite integral $\int_a^b f(x) dx$. Different situations arise, which means the analytical method we have developed so far cannot be used

to evaluate a definite integral. For example an integrand may not have an obvious antiderivative such as $\cos x^2$ or $\frac{1}{\ln x}$ or maybe the integrand is represented by an individual data points, which makes finding an antiderivative impossible. When analytical methods fail, we often turn to numerical methods [2], which are typically done on a calculator or computer.

These methods do not produce exact values of the definite integrals, but provide approximations that are generally accurate. Some of the most advanced methods for which software is widely available are rectangle rule method, trapezoidal rule method, Simpson's rule method, Newton-Cotes method, Romberg method, Gauss method and so on [3-6]. 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; Newton-Cotes method is one of the constructing integrand based on the interpolating functions, but the convergence is not guaranteed for higher order 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.

Echolocation is an important feature of bat behavior. This means, bats emit a sound pulse and listen to the echo bouncing back from obstacles whilst flying. This phenomenon has inspired Yang [7] to develop the Bat Algorithm (BA). The algorithm obtained good results dealing with lower-dimensional optimization problems, but may become problematic for higher-dimensional problems because it tends to converge very fast. This paper introduces an improved Bat Algorithm by integrating it with chaos to improve the reliability of the global optimality, and they also enhance the quality of the results.

This paper is organized as follows: after introduction, the original Bat 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.

2. THE ORIGINAL BAT ALGORITHM

Bat Algorithm has been developed by Xin-She Yang in 2010 [7]. The algorithm exploits also called echolocation of bats; bats use sonar echoes to detect and avoid obstacles. It is generally known that sound pulses are transformed to frequency which is reflected from obstacle. Bats can use time delay from emission for reflection and use it for navigation. They typically emit short and loud sound impulses and the pulse rate is usually defined as 10 to 20 times per second. After hitting and reflecting, bats transform their own pulses to useful information to gauge how far away the prey is. Bats use wavelengths, that vary from range (0.7,17) mm or inbound frequencies (20,500) kHz. By implementation, pulse frequency and pulse rates have to be defined. Pulse rate can be simply determined from range 0 to 1, where 0 meaning there is no emission and 1 meaning bats are emitting maximum (5-8). This behavior can be used to formulate the new bat algorithm. Yang [7] used three generalized rules for Bat Algorithms:

- i. All bats use echolocation to sense distance, and they also predict the difference between food/prey and background barriers in some magical way.
- ii. Bats fly randomly with velocity v_i at position x_i with a fixed frequency f_{min} , varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0,1]$, depending on the proximity of their target.
- iii. Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A_0 to a minimum constant value A_{min} .

Initialization of the bat population is performed randomly. Generating the new solutions is performed by moving virtual bats according the following equations:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (1)$$

$$v_i^t = v_i^{t-1} + (x_i^t - best)f_i, \quad (2)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (3)$$

where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution. Here X is the current global best solution which is located after comparing all the solutions among all the bats.

For the local search part, once a solution is selected among the current solutions, a new solution for each bat is generated located using random walk [8-12].

$$x_{new} = x_{old} + A_t \quad (4)$$

Where ε is the scaling factor, and A_{it} is the loudness, the loudness A_0 and the rate r_i of pulse emission have to be updated accordingly as the iterations proceed. These equations are:

$$A_i^{t+1} = \alpha A_i^t \quad (5)$$

$$r_i^{t+1} = r_0^i [1 - \exp(-\gamma t)] \quad (6)$$

Where α and γ are constants.

The basic steps of BA can be summarized as the pseudocode shown in Figure 1.



Fig. 1 Pseudo code of the bat algorithm

3. CHAOS

Generating random sequences with longer periods and good consistency is very important for easily simulating complex phenomena, sampling, numerical analysis, decision making and especially in heuristic optimization [13]. Its quality determines the reduction of storage and computation time to achieve a desired accuracy [14]. Chaos is a deterministic, random-like process found in a nonlinear, dynamical system, which is non-period, non-converging and non-bounded. Moreover, it depends on its initial condition and parameters [15-17]. Applications of chaos has several disciplines including operations research, physics, engineering, economics, biology, philosophy and computer science [18-20].

Recently chaos has been 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 [16-21]. Using chaotic sequences in Bat Algorithm can be helpful to improve the reliability of the global optimality, and they 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) [16]. 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 [22-31]. 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:

$$Y_{n+1} = \mu Y_n (1 - Y_n) \quad Y \in (0,1) \quad 0 < \mu \leq 4 \quad (7)$$

3.1.2 The Sine map

The Sine map is written as the following equation:

$$Y_{n+1} = \frac{\mu}{4} \sin(\pi Y_n) \quad Y \in (0,1) \quad 0 < \mu \leq 4 \quad (8)$$

3.1.3 Iterative chaotic map

The iterative chaotic map with infinite collapses is described as:

$$Y_{n+1} = \sin\left(\frac{\mu\pi}{Y_n}\right) \quad \mu \in (0,1) \quad (9)$$

3.1.4 Circle map

The Circle map is expressed as:

$$Y_{n+1} = Y_n + \alpha - \left(\frac{\beta}{2\pi}\right) \sin(2\pi Y_n) \text{ mod } 1 \quad (10)$$

3.1.5 Chebyshev map

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

$$Y_{n+1} = \cos(k \cos^{-1}(Y_n)) \quad Y \in (-1,1) \quad (11)$$

3.1.6 Sinusoidal map

This map can be represented by

$$Y_{n+1} = \mu Y_k^2 \sin(\pi Y_n) \quad (12)$$

3.1.7 Gauss map

The Gauss map is represented by:

$$Y_{n+1} = \begin{cases} 0 & Y_n = 0 \\ \frac{\mu}{Y_n} \text{ mod } 1 & Y_n \neq 0 \end{cases} \quad (13)$$

3.1.8 Sinus map

Sinus map is formulated as follows:

$$Y_{n+1} = 2.3(Y_n)^{2\sin(\pi Y_n)} \quad (14)$$

3.1.9 Dyadic map

Also known as the dyadic map, bit shift map, $2x \text{ mod } 1$ map, Bernoulli map, doubling map or sawtooth map. Dyadic map can be formulated by a mod function:

$$Y_{n+1} = 2Y_n \text{ mod } 1 \quad (15)$$

3.1.10 Singer map

Singer map can be written as:

$$Y_{n+1} = \mu(7.86Y_n - 23.31Y_n^2 + 28.75Y_n^3 - 13.3Y_n^4) \quad (16)$$

μ between 0.9 and 1.08



3.1.11 Tent map

This map can be defined by the following equation:

$$Y_{n+1} = \begin{cases} \mu Y_n Y_n < 0.5 \\ \mu(1 - Y_n) Y_n \geq 0.5 \end{cases} \quad (17)$$

4. THE PROPOSED ALGORITHM (IBACH) FOR SOLVING NUMERICAL INTEGRATION

Suppose that segmentation S splits an integral interval [a,b] into n-subintervals: [x0,x1],[x1,x2],...,[xk-1,xk],[xn-1,xn], where xj<xj+1 for j = 1,2,...,n-1; x0 = a, and xn= b, also define Δxk= xk - xk-1 for k = 1,2,...,n. Using this notation, the integral f(x) in [a,b] can be approximated as[32]:

$$\int_a^b f(x) \approx \sum_{k=1}^n \frac{1}{6} \left(f(x_{k-1}) + 4f\left(\frac{x_{k-1}+x_k}{2}\right) + f(x_k) \right) \Delta x_k \quad (18)$$

In the proposed chaotic Bat algorithm, we used chaotic maps to tune the Bat algorithm parameters and improve the performance [19]. The steps of the proposed chaotic Bat Algorithm for solving definite integral are as follows:

Step 1 Set the initial conditions: population xi (i = 1, 2 ...n) and Vi, pulse frequency fi at xi and pulse rates ri and the loudness Ai

Step 2 Calculate the average position and the optimal position of the bat colony.

Step 3 Using the equations 2 to 4 update velocities and locations/solutions and Generate new solutions by adjusting frequency.

$$f_i = f_{min} + (f_{max} - f_{min}) S_i, \text{ where } S_i = \text{chaotic map} \quad (19)$$

$$v_i^t = v_i^{t-1} + (x_i^t - \text{best}) f_i S_i \quad (20)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (21)$$

Step 4 If (rand > ri) then select a solution among the best solutions and generate a local solution around the selected best solution with the following equation

$$X_{new} = X_{old} + \epsilon A_t \quad (22)$$

Where ε ∈ [-1,1] If not, skip this step.

Step 5 If (rand < Ai & f(xi) < f(x)) then accept the new solutions. Increase ri and reduce Ai with the following two equations

$$A_i^{t+1} = \alpha A_i^t \quad (23)$$

$$r_i^{t+1} = r_0^i [1 - \exp(-\gamma t)] \quad (24)$$

If not, skip this step.

Step 6 Rank the bats and find the current best X.

Step 7 If the iterations attain to the maximum number, then stopped and output the global optimal solution. If not, go to step 2 to continue the search.

5. NUMERICAL RESULTS

Several examples have been done to verify the weight of the proposed algorithm. The initial parameters are set at n= 40, fmin=0, fmax= 2, A0 = (1,2), r=(0,0.1). The results of IBACH algorithm are conducted from 25 independent runs for each integrand. The selected chaotic map for all examples is the Sinusoidal map, whose equation is shown below:

$$Y_{n+1} = \cos(k \cos^{-1}(Y_n)) \quad Y \in (-1,1) \quad (25)$$

Where n is the iteration number, 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 MATLAB.

The integral values of functions x²e^{-x} in [0,2], sin²x in [0,2] and $\frac{1}{1+x^2}$ in [0,1] are selected for experiments.

Table 1 integral value of selected functions

F(x)	Monte Carlo Method	Midpoint Rule	Trapezoidal Rule	Simpson's Rule	IBACH	Exact value
x ² e ^{-x}	0.646649	0.646659	0.646633	0.646651	0.646647	0.646647



$\sin^2 x$	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
e^x	6.3892	6.3761	8.3891	6.4207	6.3891	6.3891
e^{-xz}	0.74684	0.74678	0.74621	0.74683	0.74682	0.74682

Table 1 shows the results of IBACH algorithm are privileged compared with the results of Standard Monte Carlo method, trapezoidal rule, Simpson's rule and midpoint rule. In comparison with exact values we find that the results of IBACH algorithm are very close to the exact values of selected functions under the 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 would be hard to go past the classical methods; usually though, users will choose to use IBACH algorithm to save their own time and to gain reliability.

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

6. CONCLUSIONS

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

REFERENCES

- [1] R. J. Schilling, "Applied Numerical Methods for Engineers", Brooks/Cole, 2002 .
- [2] E. Kreyszig, "Advanced Engineering Mathematics", Wiley, 8th Edition, 2000.
- [3] P. J. Davis and P. Rabinowitz, "Methods of Numerical Integration", Academic, 1975.
- [4] G. E. Forsythe, M. A. Malcolm, and C. B. Moler, Computer methods for mathematical computations: Prentice Hall Professional Technical Reference, 1977.
- [5] W. H. Press, Numerical recipes 3rd edition: The art of scientific computing: Cambridge university press, 2007.
- [6] J. Stoer, R. Bulirsch, R. Bartels, W. Gautschi, and C. Witzgall, Introduction to numerical analysis vol. 2: Springer New York, 1993.
- [7] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in Nature inspired cooperative strategies for optimization (NICSO 2010), ed: Springer, 2010, pp. 65-74.
- [8] X.-S. Yang, "Bat algorithm for multi-objective optimisation," International Journal of Bio-Inspired Computation, vol. 3, pp. 267-274, 2011.
- [9] X.-S. Yang and A. H. Gandomi, "Bat algorithm: a novel approach for global engineering optimization," Engineering Computations, vol. 29, pp. 464-483, 2012.
- [10] A. H. Gandomi, X.-S. Yang, A. H. Alavi, and S. Talatahari, "Bat algorithm for constrained optimization tasks," Neural Computing and Applications, pp. 1-17, 2013.
- [11] X.-S. Yang, Engineering optimization: an introduction with metaheuristic applications: Wiley. com, 2010.
- [12] T. C. Bora, L. d. S. Coelho, and L. Lebensztajn, "Bat-Inspired optimization approach for the brushless DC wheel motor problem," Magnetics, IEEE Transactions on, vol. 48, pp. 947-950, 2012.
- [13] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," Physical review letters, vol. 64, pp. 821-824, 1990.
- [14] D. Yang, G. Li, and G. Cheng, "On the efficiency of chaos optimization algorithms for global optimization," Chaos, Solitons & Fractals, vol. 34, pp. 1366-1375, 2007.
- [15] A. H. Gandomi, G. J. Yun, X.-S. Yang, and S. Talatahari, "Chaos-enhanced accelerated particle swarm optimization," Communications in Nonlinear Science and Numerical Simulation, 2012.
- [16] B. Alatas, "Chaotic harmony search algorithms," Applied Mathematics and Computation, vol. 216, pp. 2687-2699, 2010.
- [17] 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.



- [18] B. Alatas, "Chaotic bee colony algorithms for global numerical optimization," *Expert Systems with Applications*, vol. 37, pp. 5682-5687, 2010.
- [19] 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.
- [20] J. Mingjun and T. Huanwen, "Application of chaos in simulated annealing," *Chaos, Solitons & Fractals*, vol. 21, pp. 933-941, 2004.
- [21] 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.
- [22] 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.
- [23] R. Hilborn, "Chaos and nonlinear dynamics, 1994," ed: Oxford University Press, New York.
- [24] 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.
- [25] A. Erramilli, R. Singh, and P. Pruthi, *Modeling packet traffic with chaotic maps*: Citeseer, 1994.
- [26] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, pp. 459-467, 1976.
- [27] A. Wolf, "Quantifying chaos with Lyapunov exponents," *Chaos*, pp. 273-290, 1986.
- [28] R. L. Devaney, "An introduction to chaotic dynamical systems," 2003.
- [29] R. Barton, "Chaos and fractals," *The Mathematics Teacher*, vol. 83, pp. 524-529, 1990.
- [30] E. Ott, *Chaos in dynamical systems*: Cambridge university press, 2002.
- [31] C. Letellier, *Chaos in nature* vol. 81: World Scientific Publishing Company, 2013.
- [32] L. Qu and D. He, "Solving Numerical Integration by Particle Swarm Optimization," in *Information Computing and Applications*, ed: Springer, 2010, pp. 228-235.

