Trends in

Applied Sciences Research

Academic
Journals Inc.

A Comparative Analysis of Various Chaotic Genetic Algorithms for Multimodal Function Optimization

[1]S.K. Tiong, [2]David F.W. Yap and [1]S.P. Koh

[1]Center of System and Machine Intelligence, Universiti Tenaga Nasional (UNITEN), Kajang-Puchong Road, 43009 Kajang, Selangor Darul Ehsan, Malaysia

[2]Faculty of Electronics and Computer Engineering, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76109 Durian Tunggal, Melaka, Malaysia

Corresponding Author: David F.W. Yap, Faculty of Electronics and Computer Engineering, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76109 Durian Tunggal, Melaka, Malaysia

ABSTRACT

This study proposes a novel method of introducing chaotic induced genes into Genetic Algorithms (GA) in order to solve unimodal and multimodal mathematical test functions. The integration of chaotic elements based on logistic map into GA has significantly improved the accuracy in the aspect of the best fitness value. Simulation results show that the influence of Chaos theory does improve the optimization accuracy of the mathematical functions used.

Key words: Genetic algorithm, chaos, optimisation, multimodal function

INTRODUCTION

Genetic Algorithms (GA) was first suggested by John Holland in early 1970s. While conventional algorithms are derived from higher-order statistics or from a gradient of a singular measurement function, GA is a random search algorithm motivated by nature (Goldberg, 1989; De Castro and Zuben, 2002; Chen *et al.*, 2001). Sustaining its role as a global optimization technique, GA uses a coding scheme to act on numeric string represented by either binaries or decimals. The main strength of the evolutionary process for GA is the crossover and the mutation operator. While GA provides an efficient search method and has been used in many practical instances of optimization and decision-making problems, it tends to converge prematurely (Goldberg, 1989).

Chaos is a general phenomenon that occurs in nonlinear dynamic systems with numerous applications especially in the field of engineering and physics. Minute differences in initial conditions would cause widely diverging/divergent results for chaotic systems, so continuous predictions are impractical (Kellert, 1993). Chaotic behaviour can be seen in many natural systems such as global weather patterns (Gleick, 1988) and healthy heart rhythms (Goldberger *et al.*, 1990).

Thus, in order to enhance the performance of GA, an array of chaotic genes is injected into the GA and this method is expected to improve the accuracy of the fitness value. The functions to be used for comparing optimization results include Rastrigin's, De Jong's first, Rosenbrock's and Griewangk's function.

CHAOTIC MAPS AND GENETIC ALGORITHM

Chaotic maps: Using chaotic sequences instead of randomized number generators appears to be able to improve the performance of conventional heuristic algorithms. While the initial population

Table 1: Common chaotic maps

Map name	Equations
Logistic	$x_{n+1} = rx_n (1-x_n), r = 4$
Cubic	$x_{n+1} = rx_n (x^2_n-1)+x_n, r = 3$
Ricker's	$x_{n+1} = x_n e^{[r(1-x_n)]}, r = 20$
Hénon	$x_{n+1} = y_n+1-ax^2_n$
	$y_{n+1} = bx_n \ a = 1.4, b = 0.3$
Ikeda	$x_{n+1} = 1+\mu (x_n \cos t_n - y_n \sin t_n)$
	$y_{n+1} = \mu (x_n \sin t_n - y_n \cos t_n)$
	$t_n = 0.4 - \dfrac{6}{1 + x^2_n + y^2_n}, \mu = 0.9$
Lorenz	$\dfrac{dx}{dt} = \alpha(y - x)$
	$\dfrac{dy}{dt} = x(\gamma - z) - y$
	$\dfrac{dz}{dt} = xy - \beta z$
	$\alpha = 16, \beta = 4, \gamma = 45.92$

or genes are well-spread out by using chaotic maps, randomized genes are normally generated using Gaussian or Normal distribution. The commonly used chaotic maps are presented in Table 1. All of the chaotic maps listed in Table 1 are in real space domain. Logistic, Cubic and Ricker's (Chen and Huang, 2011) are one-dimensional maps while Hénon and Ikeda (Driebe, 1999) are in two dimensional space. The Lorenz map is a three-dimensional map and is in the continuous time domain. The rest of the chaotic maps are parameterized by a discrete-time domain.

Genetic algorithm: GA is a search technique that has a representation of the problem states (by using a set of chromosomes) and also has a set of operations to move through the search space. Each chromosome represents an individual solution (gene) to the problem. The set of individual solutions or genes forms a population. Genes in the population improve generation by generation through a set of operation that GA uses during the search process. During each generation, genes will go through the process of selection, crossover and mutation (Emmeche, 1994).

In selection, an objective function (fitness function) is used to assess the quality of the solution. The fittest solutions from each generation are kept. The crossover function generates new solutions given a set of selected members of the current population. Crossover exchanges genetic material between two single chromosome parents. Lastly, mutation would cause a sudden change in chromosomes in an unanticipated manner. As such, some values of a chromosome are altered by adding random values to the current values, therefore producing a different solution. The mutation operation is a smart technique to escape from local optimal trap in which state-space search algorithms may fall into.

Genetic algorithm with chaotic genes: The operation of a standard GA with initial chaotic genes can be represented in pseudo-code format in Fig. 1, while the summary is as follows (Goldberg, 1989):

Step 1: Generate an initial population of individuals using predefined chaotic map
Step 2: Calculate the fitness values of the individuals in the current population

```
begin GA
  c: = 0 { counter }
  Initialize population based on chaotic map
  Evaluate population { calculate fitness values }
  Do:
    c: = c+1
    Selection
    Crossover
    Mutation
    Evaluation
  end
end GA
```

Fig. 1: Pseudo-code of the standard GA

Step 3: Select individuals for reproduction
Step 4: Apply crossover and mutation operator
Step 5: Compute the fitness values of the individuals
Step 6: Select the best individuals to create the new population
Step 7: Steps 3 to 6 are repeated until a pre-defined stopping criterion is attained

In the present work, the classical GA is used with binary codification, scattered crossover, one individual elitism and stochastic uniform selection while the chaotic maps injected are based on Table 1.

MATHEMATICAL TEST FUNCTIONS

In order to compare the performance of a normal GA with injected chaotic genes, four mathematical test functions would be used (Suganthan *et al.*, 2005). These test functions provide a benchmarking method in testing the efficiency of the global optimization algorithms. For the next subsections, the details of each test function used in this study would be discussed.

Rastrigin's function: Rastrigin's function is mathematically defined as:

$$f_1(x) = \sum_{i=1}^{n} \left(x_i^2 - 10\cos\left(2\pi x_i\right) + 10 \right) \text{ where } -5.12 \leq x_i \leq 5.12 \tag{1}$$

The global minimum is located at the origin and its function value is zero. Rastrigin's function of dimension 2 is shown in Fig. 2.

De Jong's first function: De Jong's first function is mathematically defined as:

$$f_2(x) = \sum_{i=1}^{n} x_i^2 \text{ where } -5.12 \leq x_i \leq -5.12 \tag{2}$$

The global minimum is located at the origin and its function value is zero. De Jong's first function of dimension 2 is shown in Fig. 3.
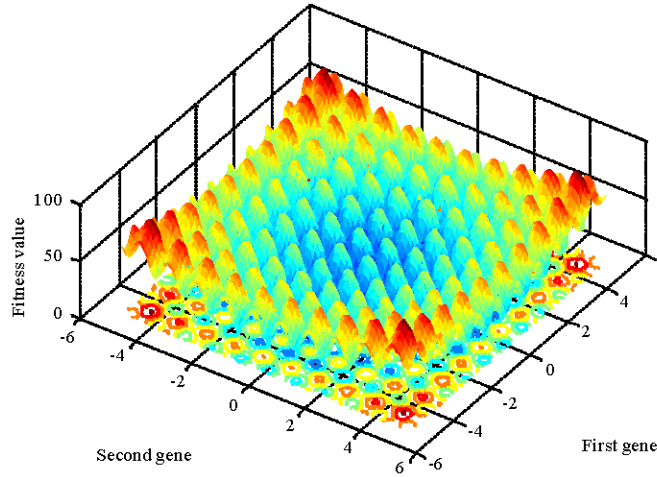
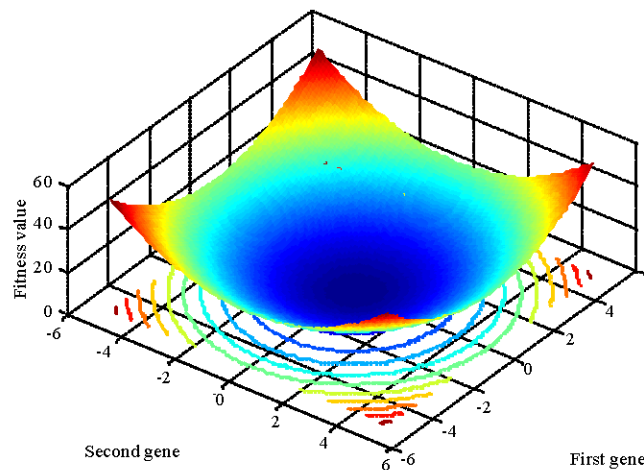Fig. 2: A two-dimensional plot of Rastrigin's function

Fig. 3: A two-dimensional plot of De Jong's first function

De Jong's second function (also known as Rosenbrock's Function): De Jong's second function is mathematically defined as:

$$f_3(x) = \sum_{i=1}^{n-1} \left(100 \left(x_{i+1} - x_i^2 \right)^2 \right) + \left(x_i - 1 \right)^2 \right) \text{ where } -2.048 \leq x_i \leq -2.048 \tag{3}$$

The global minimum is located at (1, …, 1) and its function value is zero. De Jong's second function of dimension 2 is shown in Fig. 4.

Griewangk's function: Griewangk's function is mathematically defined as:

$$f_2(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1 \text{ where } -600 \leq x_i \leq -600 \tag{4}$$
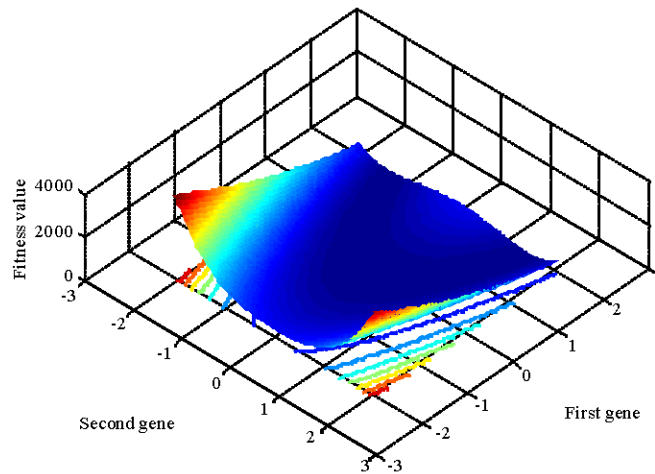
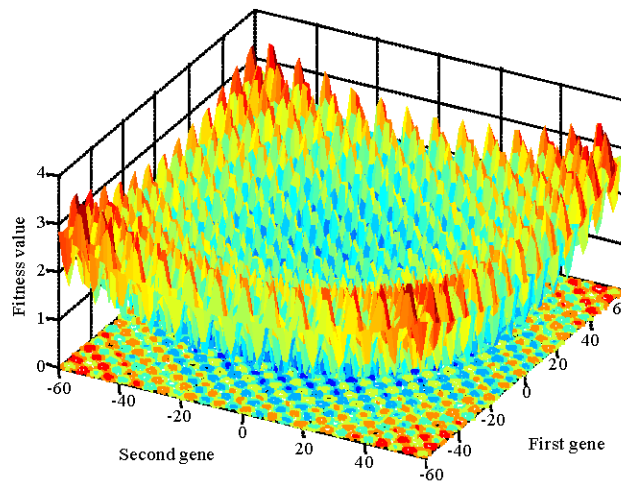Fig. 4: A two-dimensional plot of De Jong's second function

Fig. 5: A two-dimensional plot of Griewangk's function

The global minimum is located at the origin and its function value is zero. Griewangk's function of dimension 2 is shown in Fig. 5.

Simulation results and discussions: The comparison in simulation performance between GA and the injected chaotic genes was carried out by using a computer with AMD Phenom 9600B Quad-Core CPU running at 2.30 GHz, 2 GB of RAM and Windows Vista operating system. For the normal GA, the scattered crossover, the Gaussian mutation and the stochastic uniform selection operators are used. The value of crossover rate and mutation rate was set at 0.8 and 0.01, respectively. For the GA with injected chaotic maps, the parameters of the chaotic maps used are shown in Table 1. The initial population size and number of iterations are fixed at 50 for all experiments. The simulation results are shown in Table 2 with different mathematical test function. A better accuracy is achieved for fitness values closer to zero Thus, Logistic, Hénon and Ikeda function performed better than the standard GA in all of the test functions used. Similarly,

Table 2: Comparison of fitness value results using normal GA and injected chaotic maps

Map	Rastrigins	Griewangk	De Jong 1	De Jong 2
GA	7.72E-02	3.77E-04	1.12E-03	4.98E-02
Cubic	7.04E-01	1.60E-03	6.07E-08	1.49E-02
Ricker's	8.52E-01	1.58E-03	4.04E-03	2.41E-01
Logistics	1.20E-05	3.04E-08	6.07E-08	3.83E-07
Hénon	3.38E-03	2.73E-07	1.71E-05	2.67E-02
Ikeda	6.77E-03	1.71E-05	1.74E-05	3.88E-04
Lorentz flow	7.12E-01	2.21E-03	4.59E-03	5.87E-02

there is an improvement for Cubic function in comparison with the standard GA for De Jong 1 and De Jong 2 test function, respectively. The improvement in the accuracy of the fitness value is probably due to the randomness of the genes based on the chaotic map that covers a wide range of possible solutions, especially in the initialization stage of the GA. Overall, GA with Logistics chaotic map performed the best in all the four mathematical test functions with the lowest fitness value in the order of E-08.

A comparison with the classical particle swarm optimization, genetic algorithm and artificial immune system by other authors (Edge *et al.*, 2006; Gu *et al.*, 2009; Valdez and Melin, 2008; Vieira *et al.*, 2009; Yap *et al.*, 2009) shows that the chaotic GA proposed here is superior in terms of achieving the fitness value closest to zero. In addition, in Ma and Li (2009), Ru and Jianhua (2008), Sun *et al.* (2009), Yap *et al.* (2011a, b), the performance of chaotic GA is on par with other hybrid immune based algorithms with fitness values in the region of E-06 to E-08. With the increase in accuracy by injecting chaotic maps into normal GA; applications such as scheduling problems, daily rostering systems or other optimization problems would bear better and accurate results.

CONCLUSION

The simulation results of GA and chaotic maps was presented and compared. The usage of the multimodal test function shows that the GA and chaotic maps have comparable fitness value with differences in the order of thousands or less from the global minimum. From the results, Logistic, Hénon and Ikeda performed better than the standard GA in obtaining the lowest fitness value of the multimodal test functions. Future work may include improved or modified GA and chaotic map with other Particle Swarm Optimization or Artificial Immune System as a hybrid algorithm that will be used to optimize other complicated test function.

REFERENCES

Chen, G., X. Wang, Z. Zhuang and D. Wang, 2001. Genetic Algorithm and Application. Post and Telecom Press, Beijing.

Chen, G. and Y. Huang, 2011. Chaotic Maps: Dynamics, Fractals and Rapid Fluctuations, Synthesis Lectures on Mathematics and Statistics. Morgan and Claypool Publishers, Texas.

De Castro, L.N. and F.J. Von Zuben, 2002. Learning and optimization using clonal selection principle. IEEE Trans. Evolutionary Comput., 6: 239-251.

Driebe, D.J., 1999. Fully Chaotic Maps and Broken Time Symmetry. Vol. 4, Springer, Boston, ISBN: 9780792355649, Pages: 164.

Edge, K.S., G.B. Lamont and R.A. Raines, 2006. A retrovirus inspired algorithm for virus detection and optimization. Proceedings of Genetic and Evolutionary Computation Conference, July 8-12, 2006, Seattle, Washington, pp: 103-110.

Emmeche, C., 1994. The Garden in the Machine: The Emerging Science of Artificial Life. Princeton University Press, New Jersey.

Gleick, J., 1988. Chaos: Making a New Science. 1st Edn., Penguin Books, New York, ISBN-10: 0140092501, pp: 352.

Goldberg, D.E., 1989. Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley, New York, USA.

Goldberger, A.L., R.R. David and J.W. Bruce, 1990. Chaos and fractals in human physiology. Sci. Am., 262: 42-49.

Gu, J., L. Lin and H. Wang, 2009. A hybrid particle swarm optimization algorithm for multimodal function optimization. Proceedings of the International Workshop on Intelligent Systems and Applications, May 23-24, 2009, Wuhan, pp: 1-4.

Kellert, S.H., 1993. In the Wake of Chaos: Unpredictable Order in Dynamical Systems. University of Chicago Press, Chicago, IL., USA., ISBN-13: 9780226429748, pp: 32.

Ma, J.H. and L. Li, 2009. Well logging automatic core relocation based of the immune particle swarm opimization algorithm. Proceedings of the International Conference on Computational Intelligence and Software Engineering, December 11-13, 2009, Wuhan, pp: 1-7.

Ru, N. and Y. Jianhua, 2008. A GA and particle swarm optimization based hybrid algorithm. Proceedings of the Congress on Evolutionary Computation, June 1-6, 2008, Hong Kong, pp: 1047-1050.

Suganthan, P.N., N. Hansen, J.J. Liang, K. Deb, Y.P. Chen, A. Auger and S. Tiwari, 2005. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical Report, Nanyang Technological University, Singapore and IIT Kanpur, India, pp: 1-50.

Sun, Y., W. Zhang and M. Zhang, 2009. Modified particle swarm optimization based on immune clone principle for analog-matching of equivalent system. Proceedings of International Conference on Automation and Logistics, August 5-7, 2009, Sheyang, China, pp: 1135-1138.

Valdez, F. and P. Melin, 2008. Comparative study of particle swarm optimization and genetic algorithms for complex mathematical functions. J. Automation Mobile Robotics Intell. Syst., 2: 43-51.

Vieira, I.N., A.J.M. Silva, B.S.L.P de Lima and B. Jacob, 2009. A comparative study applied to risers optimization using bio-inspired algorithms. Int. J. Model. Simulat. Petroleum Ind., 3: 5-12.

Yap, D.F.W., S.P. Koh and S.K. Tiong, 2009. A comparative analysis on the performance of particle swarm optimization and artificial immune systems for mathematical test functions. Aust. J. Basic Applied Sci., 3: 4344-4350.

Yap, D.F.W., S.P. Koh, S.K. Tiong and S.K. Prajindra, 2011a. A swarm based artificial immune systems for solving multimodal function. Applied Artificial Intelli., 25: 693-707.

Yap, D.F.W., S.P. Koh, S.K. Tiong and S.K. Prajindra, 2011b. Particle swarm based artificial immune system for multimodal function optimization and engineering application problem. Trends Applied Sci. Res., 6: 282-293.