

Particle Swarm based Artificial Immune System for Multimodal Function Optimization and Engineering Application Problem

¹D.F.W. Yap, ²S.P. Koh, ²S.K. Tiong and ²S.K. Prajindra

¹Faculty of Electronics and Computer Engineering, Universiti Teknikal Malaysia Melaka (UTeM), Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

²College of Engineering, Universiti Tenaga Nasional (UNITEN), km 7, Kajang-Puchong Road, 43009 Kajang, Selangor Darul Ehsan, Malaysia

Corresponding Author: David F.W. Yap, Faculty of Electronics and Computer Engineering, Universiti Teknikal Malaysia Melaka (UTeM) Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia Tel: +606 555 2094 Fax: +606 555 2112

ABSTRACT

Artificial Immune Systems (AIS) has generated great interest among researchers as the algorithm is able to improve local searching ability and efficiency. However, the rate of convergence for AIS in finding the global minima is rather slow as compare to other Evolutionary Algorithms. Alternatively, Genetic Algorithms (GAs) and Particle Swarm Optimization (PSO) have been used effectively in solving complicated optimization problems, but they tend to converge prematurely at the local minima. In this study, the Hybrid AIS (HAIS) is proposed by combining the good features of AIS and PSO in order to reduce this shortcoming. By comparing the optimization results of the mathematical functions and the engineering problem using GA, AIS and HAIS, it is observed that HAIS achieved better performances in terms of accuracy, convergence rate and stability.

Key words: Clonal selection, affinity maturation, mutation, antibody, antigen

INTRODUCTION

Various optimization methods and algorithms have been developed over the past few years for solving real-world optimization problems. Some of the most well-known optimization algorithms stated here are Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 2001), Artificial Immune Systems (AIS) (De Castro and Zuben, 2002) and Genetic Algorithms (GA) (Chen *et al.*, 2001). PSO, AIS and GA are widely used for solving complex optimization problems but each method has strengths and restrictions.

GA was first suggested by John Holland in early 1970s. While conventional algorithms are derived from higher order statistics or gradient of a singular measurement function, GA is a random searching algorithm motivated by nature (Chen *et al.*, 2001). Sustaining its role as a global optimization technique, it uses a coding scheme to act on numerical string represented by either binaries or decimals. The main strength of the evolutionary process for the GA is the crossover operator and mutation operator. Furthermore, GA provides an efficient search method and has been used in many practical instances of optimization and decision-making problems (Michalewicz, 1996). Conversely, GA is prone to premature occurrence that results in convergence at the local optima. This weakness is generally caused by the poor diversity of the population.

The PSO make use of social behaviour (Kennedy and Eberhart, 2001; Pant *et al.*, 2008) of individuals living together in groups. Each individual tries to imitate other better group members in order to improve itself. That way, optimization procedure is being carried out by the group members as described by Pant *et al.* (2008). The performance of PSO depends on the way the particle move in the search space with a velocity that is constantly updated. Much to the amazement of the research community, PSO has very few parameters and values to tweak, which makes it exceptionally easy to implement and has higher speed of convergence. Nevertheless, as reported by Angeline (1998) and Riget and Vesterstrøm (2002), the performance of PSO degrades when the numbers of generations are increased as it tends to converge prematurely given the aptitude constraint of the model and the diminished diversity of the search space.

Alternatively, AIS, a new branch of computational intelligence, is a new intelligent problem-solving technique which finds applications in optimization, computer security, data clustering, pattern recognition or even fault tolerance (Dasgupta, 2006). The natural immune system uses a diversity of evolutionary and adaptive techniques to protect organisms from foreign pathogens and misbehaving cells in the body (De Castro and Zuben, 2005). AIS seek to capture some aspects of the natural immune system in a computational framework for solving engineering problems (Glickman *et al.*, 2005). Moreover, immune based applications are based on immune memory. Thus, the diversity during the process of evolution is preserved and the convergence of global optimization is assured. While, the convergence to the global optima is ascertained, the rate and speed of convergence is relatively slow and should be improved. Clonal selection algorithm that is used here in the simulation uses the principles of clonal expansion and affinity maturation as the main forces of the evolutionary process (De Castro and Timmis, 2002a, b).

As a result, an improved AIS algorithm based on the PSO is established in this study to improve the performance of diversity and convergence. This new algorithm, HAIS (Hybrid AIS), is expected to enhance global search ability and to prevent itself from being trapped in the local optimum when small population size is used. In addition, the ease of implementation attribute is sustained in the proposed algorithm.

GENETIC ALGORITHM METHODOLOGY

Genetic Algorithm (GA) is a search technique that has a representation of the problem states (represented using a set of chromosomes) and also has a set of operations to move through the search space. Each chromosome represents an individual solution (genes) to the problem. The set of individual solutions or genes forms a population. Genes in the population are improved across generation through a set of operation that GA uses during the search process. During each generation, the 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 function of crossover 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 optima trap in which state-space search algorithms may fall into. The operation of a standard GA is summarized as follows (Michalewicz, 1996):

- Step 1:** Generate a random initial population of individuals
- Step 2:** Calculate the fitness values of the individuals in the current population
- 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 study, the classical GA is used, with binary codification, scattered crossover, one individual elitism and stochastic uniform selection. Initial population size and number of iterations are fixed at 50.

PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) was originally proposed by Kennedy and Eberhart (1995) as a simulation of social behaviour of organisms that live and interact within large groups. The essence of PSO is that particles move through the search space with velocities which are dynamically adjusted according to their historical behaviours. This mimics the social behaviour of animals such as bird flocking and fish schooling. With several alterations on the original model, the algorithm could optimize complex functions based on the concept of Swarm Intelligence methods (Kennedy and Eberhart, 2001).

PSO algorithm starts with a group of random particles that searches for optima by updating each generation. Each particle is represented by a volume-less particle in the n-dimensional search space. The *i*th particle is denoted as $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$. At each generation, each particle is updated by ensuring two best values. They are the best solution that has been achieved (mbest) and the global best value (gbest) that is obtained so far by any particle in the population.

With the inclusion of the inertia factor, ω , the equations are (Shi and Eberhart, 1998);

$$v_{i+1} = v_i \omega + \alpha_1 \cdot \text{rnd}() \cdot (\text{mbest}_i - x_i) + \alpha_2 \cdot \text{rnd}() \cdot (\text{gbest}_i - x_i) \quad (1)$$

$$x_{i+1} = x_i + v_i \quad (2)$$

where, $\text{rnd}()$ is a random number independently generated within the range [0,1] and α_1 and α_2 are two learning factors which control the influence of the individual's knowledge and that of the neighbourhood respectively. The PSO optimization algorithm can be written as follows:

Generate a random initial swarm of particles, assigning each one a random position and velocity.

- Step 1:** Compute the fitness values of the N particles
- Step 2:** Update the values of the best position of each particle and the best position found by the swarm
- Step 3:** Update the position and the velocity of every particle according to Eq. 1 and 2
- Step 4:** Steps 2 to 4 are repeated until a pre-defined stopping condition is reached

ARTIFICIAL IMMUNE SYSTEM-CLONAL SELECTION FOR OPTIMIZATION

Artificial Immune Systems (AIS) are inspired by theoretical immunology and observed immune functions, principles and models, which are applied to engineering problem solving (De Castro and Timmis, 2002a, b). The clonal selection algorithm is a branch of AIS with principles extracted from clonal expansion and affinity maturation (De Castro and Zuben, 2002). The clonal selection theory explains that when an antigen (Ag) is detected, antibodies (Ab) that best recognize this Ag will proliferate by cloning. This immune response is specific to each Ag.

The immune cells will reproduce in tandem with a replicating Ag until it is successful in recognizing and fighting against this Ag. Some of the new cloned cells will be differentiated into plasma cells and memory cells. The plasma cells produce Ab and will undergo mutation that will promote their genetic variation. The memory cells are responsible for the immunologic response for future Ag invasion. Subsequently, the selection mechanism will keep the cells with the best affinity to the Ag in the next population. The process of a standard clonal selection algorithm is described as (De Castro and Timmis, 2002a, b):

- Step 1:** Generate a random initial population of Ab
- Step 2:** Compute the fitness of each Ab
- Step 3:** Generate clones by cloning all cells in the Ab population
- Step 4:** Mutate the clone population to produce a mature clone population
- Step 5:** Evaluate affinity values of the clones' population
- Step 6:** Select the best Ab to compose the new Ab population
- Step 7:** Steps 3 to 6 are repeated until a pre-defined stopping condition is reached

Hybrid artificial immune system: The good traits of the PSO especially in terms of convergence speed and coding simplicity are incorporated into AIS. Alternatively, AIS is able to prevent the population from being trapped in the local optimum. The process of the HAIS can be characterized in pseudo-code format in Fig. 1.

The working of HAIS is explained below:

- Step 1:** Select the best particles, N_1 (consisting of $N/2$ particles) from PSO
- Step 2:** Generate a random initial population of Ab, N_2 (consisting of the other half, $N/2$ particles and regard every particle as an antibody)
- Step 3:** Combine N_1 with N_2 and compute the fitness of each Ab
- Step 4:** Generate clones by cloning all cells in the Ab population
- Step 5:** Mutate the clone population to produce a mature clone population
- Step 6:** Evaluate affinity values of the clones' population
- Step 7:** Select the best Ab to compose the new Ab population
- Step 8:** Steps 4 to 7 are repeated until a pre-defined stopping condition is reached

MATHEMATICAL FUNCTIONS AND ENGINEERING PROBLEM

In order to compare the performance of GA, AIS and HAIS, four mathematical test functions would be used (Suganthan *et al.*, 2005). These test functions provide a firm benchmarking method in testing the efficiency of the global optimization algorithms. For the next subsections, the details of each of the test functions used in this study would be discussed. In addition, an engineering problem is presented at the end of the subsection.

```

begin HAIS

//Computing N1 from PSO
c:=0 { counter }
Initialize particle
Do:      {for N1 particles}
Calculate fitness value
If the fitness value is better than the best fitness value (mbest) in history
Set current value as the new mbest
Choose the particle with the best fitness value of all the particles as the gbest
Calculate and update particle information using Eq. 1 and 2
c:=c+1
end
Select N1 particles with best fitness value

//AIS evaluation
d:=0 { counter }
Initialize population {consisting N2 particles}
Combine N1 with N2
Do:
Compute affinity
Generate clones
Mutate clones
Replace lowest affinity Ab with a new randomly generated Ab
d:=d+1
end

end HAIS
    
```

Fig. 1: Pseudo-code of the HAIS Algorithm

Rastrigin’s function: Rastrigin’s function is mathematically defined as in Eq. 3:

$$f_1(x) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10) \tag{3}$$

where, $-5.12 \leq x_i \leq 5.12$. The global minimum is located at the origin and its function value is zero. The two-dimensional Rastrigin’s function is shown in Fig. 2.

Griewangk’s function: Griewangk’s function is mathematically defined as in Eq. 4:

$$f_2(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{4}$$

where, $-600 \leq x_i \leq 600$. The global minimum is located at the origin and its function value is zero. The two-dimensional Griewangk’s function is shown in Fig. 3.

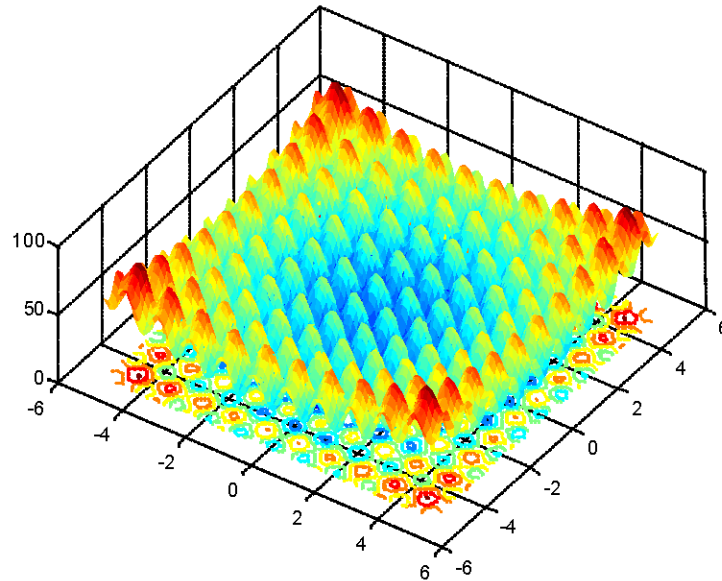


Fig. 2: A two-dimensional plot of Rastrigin's function

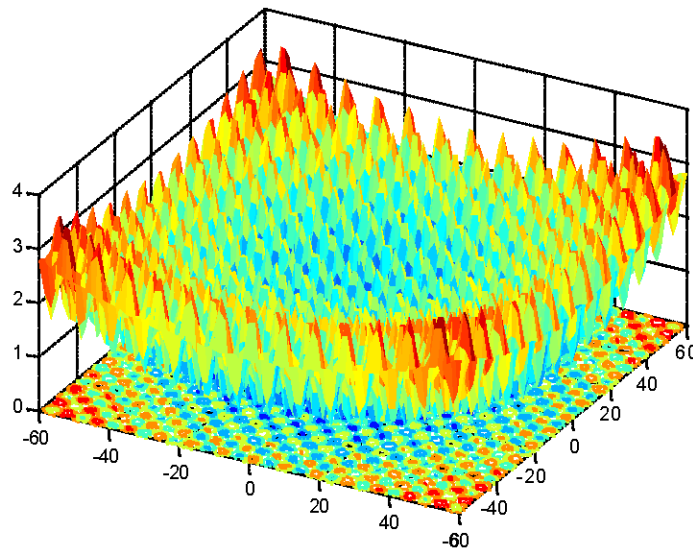


Fig. 3: A two-dimensional plot of Griewangk's function

Rosenbrock's function: Rosenbrock's function is mathematically defined as in Eq. 5:

$$f_3(x) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 \right) + (x_i - 1)^2 \quad (5)$$

where, $-2.048 \leq x_i \leq 2.048$. The global minimum is located at (1,1) and its function value is zero. The two-dimensional Rosenbrock's function is shown in Fig. 4.

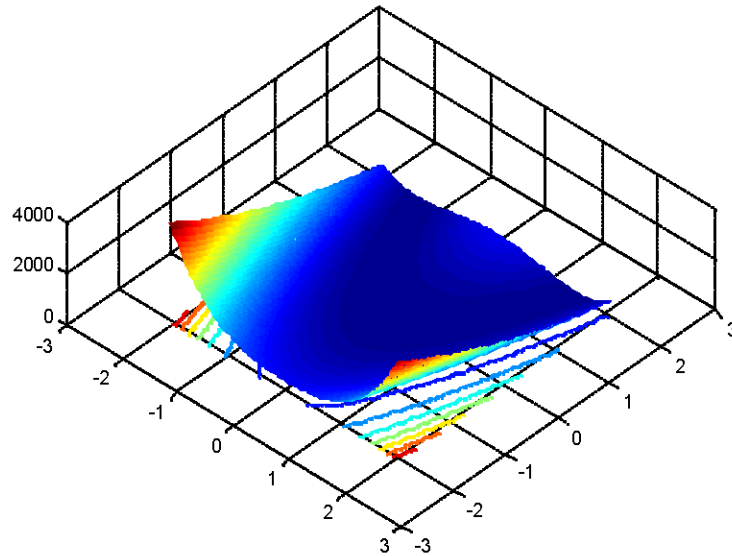


Fig. 4: A two-dimensional plot of Rosenbrock's function

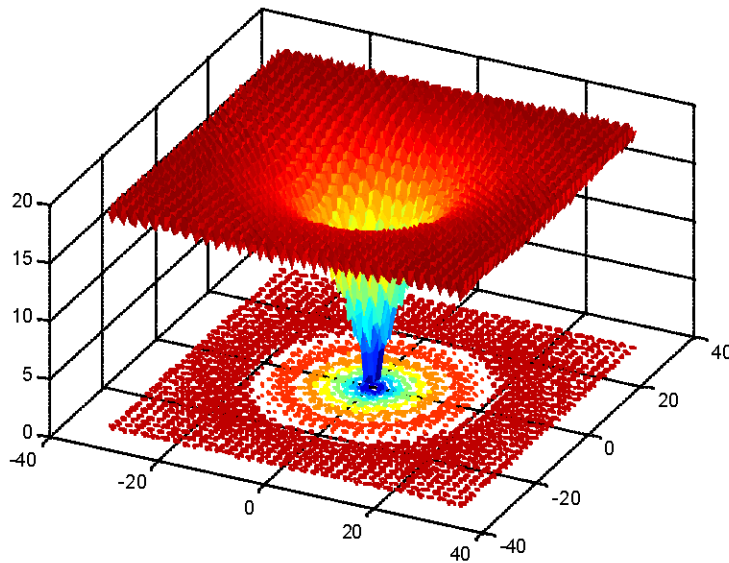


Fig. 5: A two-dimensional plot of Ackley's function

Ackley's function: Ackley's function is mathematically defined as in Eq. 6:

$$f_4(x) = 20 + e - 20e^{-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}} - e^{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)} \quad (6)$$

where, $-32.768 \leq x_i \leq 32.768$. The global minimum is located at the origin and its function value is zero. The two-dimensional Ackley's function is shown in Fig. 5.

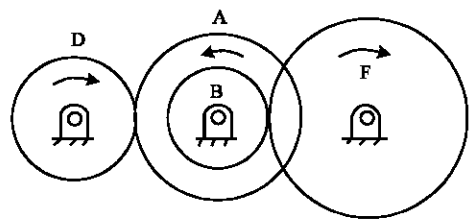


Fig. 6: Compound gear train

Engineering problem: design of a gear train: The compound gear train problem (Kannan and Kramer, 1994) shown in Fig. 6 was presented by Sandgren (1988). It is to be designed such that the gear ratio is as close as possible to $1/6.931$. Each gear must consist of between 12 and 60 number of teeth. The variables must be integers, since the number of teeth is to be an integer. The mathematical model is stated as in Eq. 7:

$$F(x) = \left\{ \frac{1}{1.6931} - \frac{x_1 x_2}{x_3 x_4} \right\}^2 \quad (7)$$

where, x_1 , x_2 , x_3 and x_4 are the teeth of the gears and $12 \leq x_i \leq 60$.

RESULTS AND DISCUSSION

The comparison in simulation performance between GA, AIS and HAIS was carried out towards the end of March 2010 using a computer in the Computer Lab., with AMD Phenom 9600 B Quad-Core CPU running at 2.30 GHz, 2 GB of RAM and Windows Vista Enterprise operating system. For both the HAIS and AIS, which were based on clonal selection algorithm (De Castro and Zuben, 2002), mutation probability is 0.01, clone size factor is 2, while initial population size and number of iterations fixed at 50. The simulation results in terms of best fitness value, mean best fitness and standard deviation are shown in Table 1, whereas the performance of GA, AIS and HAIS are compared and shown in Fig. 7-10 for each of the fitness function used.

From Fig. 7-10, GA and HAIS converged faster than AIS while HAIS gave better fitness value which is closer to the global minimum among the three. The increased in HAIS convergence and accuracy clearly show the benefit of injecting PSO qualities to improve performance. The HAIS was found to be better in obtaining the lowest fitness value of the functions of Eq. 3-5 as compare to (Edge *et al.*, 2006). The study done by Valdez and Melin (2008) clearly showed that HAIS achieved lower fitness value for the Ackley's function and Griewank's function compare to the standard GA and PSO algorithms.

Another hybrid algorithm based on Clonal Selection was proposed by Mitra and Venayagamoorthy (2008). However, the HAIS outperformed their results for the Rosenbrock's function and Rastrigin's function. Same goes to the improved PSO algorithm suggested by Pant *et al.* (2008), where HAIS performed better for the Rastrigin's function and Griewank's function.

The engineering problem simulation was executed using the same parameters as in the mathematical function simulation. Table 2 shows the results for the Gear Train Design. As expected, HAIS performs better in terms of accuracy compare to GA and AIS.

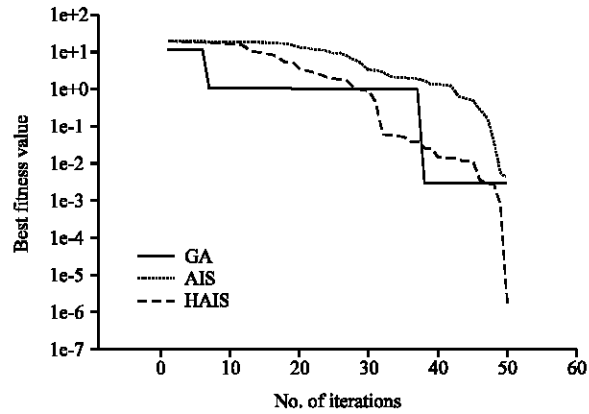


Fig. 7: Fitness value comparison for Rastrigin's function

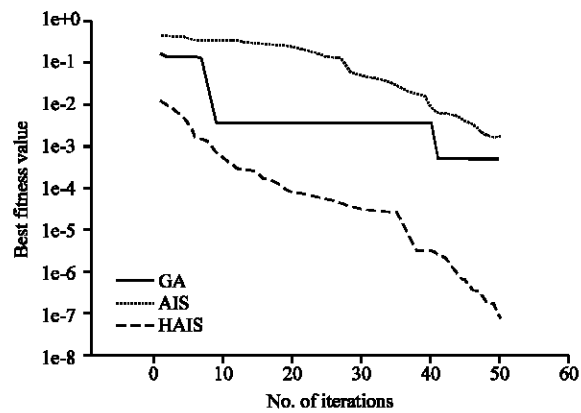


Fig. 8: Fitness value comparison for Griewangk's function

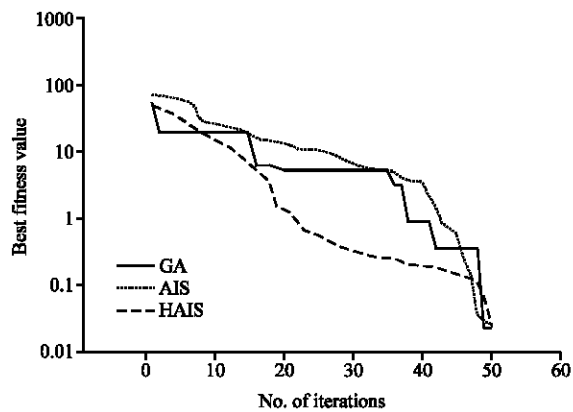


Fig. 9: Fitness value comparison for Rosenbrock's function

Table 1: Comparison of fitness value results using GA, AIS and HAIS

Fitness function	GA			AIS			HAIS		
	Best fitness value	Mean	SD	Best fitness value	Mean	SD	Best fitness value	Mean	SD
Rastrigin	0.0029486	5.73427	3.55734	0.0043441	10.09287	7.94867	1.3009e-006	9.77215	7.20046
Griewangk	0.00049315	0.085547	0.047226	0.0017484	0.21997	0.15363	7.5765e-008	0.00595	0.002494
Rosenbrock	0.0231	27.82825	10.04669	0.0258	36.0741	20.72595	0.0239	24.8636	13.93407
Ackley	0.5750	1.9742	0.70956	0.6242	2.63105	0.97952	2.3891e-003	2.28114	1.39911

Table 2: Best gear train design results obtained from the simulation

Parameters	GA	AIS	HAIS
x_1	12	14	18
x_2	20	31	18
x_3	32	60	42
x_4	52	56	53
F(x)	4.86e-005	0.00022838	1.6211e-006

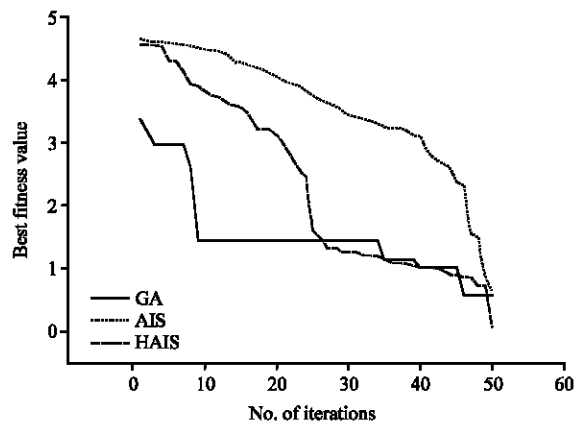


Fig. 10: Fitness value comparison for Ackley's function

CONCLUSION

The classical PSO is stochastic in nature and simple to implement. But as the search goes on, the population diversity becomes worst which in turn leads to premature convergence. Even GA is also affected by premature convergence at the local optima. Thus, in order to minimize this problem, a hybrid AIS was presented in this study to combine the good qualities of PSO with the properties of AIS. The simulation results of GA, AIS and HAIS was presented and compared. In contrast to GA and AIS on four multimodal test functions and an engineering problem, HAIS performed better in terms of accuracy, convergence rate, stability and robustness. Future work may include comparison with other evolutionary algorithms and on more complicated test function.

ACKNOWLEDGMENT

This study was supported in part by grant number FRGS/2010/FKEKK/TK03/1-F0085 from the Ministry of Higher Education of Malaysia.

REFERENCES

- Angeline, P.J., 1998. Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences. In: *Evolutionary Programming*, Porto, V.W., N. Saravanan, D. Waagen and A.E. Eiben, (Eds.). Vol. 7, Springer-Verlag, Berlin, ISBN: 978-3-540-64891-8, PP: 601-610.
- Chen, G., X. Wang, Z. Zhuang and D. Wang, 2001. *Genetic Algorithm and Application*. Post and Telecom Press, Beijing.
- Dasgupta, D., 2006. Advances in artificial immune systems. *IEEE Comput. Intel. Magazine*, 1: 40-49.
- De Castro, L.N. and J. Timmis, 2002a. *Artificial Immune Systems: A New Computational Approach*. Springer-Verlag, New York.
- De Castro, L.N. and J. Timmis, 2002b. *Artificial Immune Systems: A Novel Paradigm to Pattern Recognition*. In: *Artificial Neural Networks in Pattern Recognition*, Corchado, J.M., L. Alonso and C. Fyfe (Eds.). University of Paisley, UK, pp: 67-84.
- De Castro, L.N. and J.V. Zuben, 2002. Learning and optimization using clonal selection principle. *IEEE Trans. Evolutionary Comput.*, 6: 239-251.
- De Castro, L.N. and J.V. Zuben, 2005. *Recent Developments in Biologically Inspired Computing*. Idea Group Inc. Publishing, Hershy, PA, USA.
- 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, Seattle, Washington, pp: 103-110.
- Emmeche, C., 1994. *Garden in the Machine: The Emerging Science of Artificial Life*. Princeton University Press, New Jersey, USA.
- Glickman, M., J. Balthrop and S. Forrest, 2005. A machine learning evaluation of an artificial immune system. *Evol. Comput. J.*, 13: 179-212.
- Kannan, B.K. and S.N. Kramer, 1994. An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *J. Mech. Design*, 116: 405-411.
- Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization. *Proc. IEEE Int. Conf. Neural Networks*, 4: 1942-1948.
- Kennedy, J. and R. Eberhart, 2001. *Swarm Intelligence*. 1st Edn., Academic Press, San Diego, CA., ISBN: 1-55860-595-9.
- Michalewicz, Z., 1996. *Genetic Algorithms + Data Structures = Evolution Programs*. 3rd Edn., Springer, New Mexico.
- Mitra, P. and G.K. Venayagamoorthy, 2008. Empirical study of a hybrid algorithm based on clonal selection and small population based PSO. *Proceedings of IEEE Swarm Intelligence Symposium*, Sept. 21-23, USA., pp: 1-7.
- Pant, M., R. Thangaraj and A. Abraham, 2008. Particle swarm based meta-heuristics for function optimization and engineering applications. *Proceedings of the 2008 7th Computer Information Systems and Industrial Management Applications*, June 26-28, Washington, DC, USA., pp: 84-90.
- Riget, J. and J.S. Vesterstrøm, 2002. A diversity-guided particle swarm optimizer-the ARPSO. Technical Report, EVALife, Dept. of Computer Science, University of Aarhus.

- Sandgren, E., 1988. Nonlinear integer and discrete programming in mechanical design. Proceeding of the ASME Design Technology Conference, Sept. 25, Kissimmee, Florida, pp: 95-105.
- Shi, Y. and R. Eberhart, 1998. A modified particle swarm optimizer. Proceedings of the IEEE Congress on Evolutionary Computation, May 4-9, Piscataway, New Jersey, pp: 69-73.
- 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 2005, Nanyang Technological University, Singapore and IIT Kanpur, India.
- 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.