AN IMPROVED ARTIFICIAL IMMUNE SYSTEM BASED ON ANTIBODY REMINDER METHOD FOR MATHEMATICAL FUNCTION OPTIMIZATION

DAVID F.W. YAP
A. HABIBULLAH
S.P. KOH
S.K. TIONG

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

An Improved Artificial Immune System based on Antibody Remainder method for Mathematical Function Optimization

David F. W. Yap*, A. Habibullah
Faculty of Electronic and Computer Engineering,
Universiti Teknikal Malaysia Melaka (UTeM),
Hang Tuah Jaya,76100 Durian Tunggal, Melaka, Malaysia.
*david.yap@utem.edu.my

S. P. Koh, S.K.Tiong
College of Engineering,
Universiti Tenaga Nasional (UNITEN),
Km 7, Kajang-Puchong Road, 43009 Kajang, Selangor
Darul Ehsan, Malaysia.

Abstract—**Artificial immune system (AIS) is one of the nature-inspired algorithm for optimization problem. In AIS, clonal selection algorithm (CSA) is able to improve global searching ability. However, the CSA convergence and accuracy can be improved further because the hypermutation in CSA itself cannot always guarantee a better solution. Alternatively, Genetic Algorithms (GAs) and Particle Swarm Optimization (PSO) have been used efficiently in solving complex optimization problems, but they have a tendency to converge prematurely. In this study, the CSA is modified using the best solutions for each exposure (iteration) namely Remainder-CSA. The results show that the proposed algorithm is able to improve the conventional CSA in terms of accuracy and stability for single objective functions.**

Keywords-component: clonal selection, antibody, antigen, affinity maturation, mutation.

I. INTRODUCTION

Optimization problem has been a challenge to many researchers in order to find the best local searching method. This problem also leads to a branch of knowledge which is the evolutionary computing. The methods were greatly influenced by nature. Few decades ago, many methods have been developed, for instance, GA, PSO, Ant Colony or Artificial Immune System (AIS). In this study, the improved CSA is evaluated in comparison to conventional CSA and other evolutionary algorithms such as PSO and GA. These algorithms are described in the following paragraphs.

PSO was originally proposed by Kennedy and Eberhart [1]. The PSO is inspired from social behavior of individual organisms living together in groups [2]. Each individual in a group imitates other groups that are better, in order to improve its own group.

GA is inspired from a set of chromosome where each chromosome represents an individual solution (genes). The GA uses a search technique where genes in the population are improved across generation through a set of operation. During each generation, the genes go through the process of selection, cross-over and mutation [3].

AIS is one of the nature-inspired approach in solving the optimization problem. The AIS is greatly reinforced by the immune system of a living organism such as human and animal. In humans, the immune system is responsible in maintaining stability of the physiological system such as protection from pathogens. In AIS, CSA is able to improve global searching ability as it uses the principle of clonal expansion and affinity maturation as the main forces of the evolutionary process [4]

In this research, the CSA is studied to improve the performance of diversity and convergence that are responsible in finding the global solution of single objective function. There are two proposed algorithms i.e. half best insertion (HBI) CSA and single best remainder (SBR) CSA. The ease of its implementation is sustained in the proposed algorithms.

II. PSO, GA AND AIS ALGORITHM

Particle Swarm Optimization

The PSO algorithm starts with a group of random particles that searches for optimum value for each updated generation. The i_{th} particle is denoted as $X_i = (x_{i1}, x_{i2}, x_{i3}, ..., x_{in})$. During generation updating, each particle is updated by ensuing two best values. These values are the best solution (*mbest*) and the global best value (*gbest*) that has been obtained by particles in the population at particular generation. With the inclusion and inertia factor ω, the velocity equations are shown in Eqs. (1) and (2).

$$v_{i+1} = v_i\omega + \alpha_1 \cdot rnd() \cdot (mbest_i - x_i) + \alpha_2 \cdot rnd() \cdot (gbest_i - x_i) \quad (1)$$

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

Where *rnd()* is a random number between 0 and 1, α_1 and α_2 are learning factors to control the knowledge and the neighbourhood of each individual respectively. The PSO algorithm is described in the following steps.

Step	Process
1	Generate initial random particle swarms assigned with its random position and velocity
2	Compute the fittest value of *N* particles according to fitness function
3	Update values of the best position of each particle and the swarm

Step	Process
4	Update the position and velocity for each particle according to equation 1 and 2.
5	Repeat steps 3 and 4 until pre-defined stopping condition is achieved

Genetic Algorithm

As described earlier, GA uses three main processes i.e. selection, crossover and mutation to improve genes through each generation. The selection process uses the objective function to assess the quality of the solution. Then, the fittest solutions from each generation are kept. Then, the function of crossover generates new solutions given a set of selected members of the current population. In the crossover process, genetic material between two single chromosome parents is exchanged. Then, mutation triggers sudden change in the chromosomes unexpectedly. However, the mutation process is expected to avoid genes from trapping in local minima by adding random variables. The GA algorithm is described in the following steps.

Step	Process
1	Generate initial random population of individuals
2	Compute the fittest value of each individual in the current population
3	Select individuals for reproduction
4	Apply crossover and mutation operators
5	Compute the fittest value of each individual
6	Select the best individuals to generate new population
7	Repeat steps 3 to 6 until pre-defined stopping condition is achieved

Artificial Immune System

The biological immune system has been modeled into AIS for engineering application. In AIS, CSA was inspired from the fact that only antibodies (Abs) that are able to recognize antigens (Ags) are selected to proliferate. The selected Abs then enters the affinity maturation process. The algorithm was verified to be capable to solve complex problem such as multi-modal and combinatorial optimization [5].

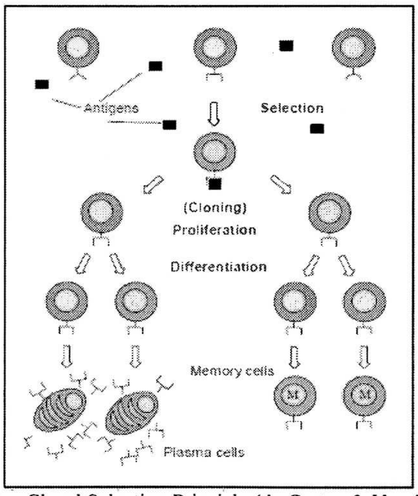

Figure 1. Clonal Selection Principle (de Castro & Von Zuben, 2001a)

The clonal selection theory describes how Abs detects the Ags and proliferate by cloning. As shown in Fig. 1, the immune cells will reproduce against the Ags. The new cloned cells are then differentiated into plasma cells and memory cells. The plasma cells produce Abs and go through mutation process to promote genetic variation. The memory cells are responsible for future Ags invasion. Finally, the selection mechanism keep the Abs with the best affinity to the Ags in the next population [4]. The CSA pseudocode is described in the following steps.

Step	Process
1	Generate an initial random population of antibodies, Abs
2	Compute the fittest value of each Ab according to fitness function
3	Generate clones by clonning all cells in the Ab population
4	Mutate the clone population to produce a mature clone population
5	Evaluate the affinitiy value for each clone population
6	Select the best Ab to compose the new Ab population
7	Repeat steps 2 to 6 until a pre-defined stopping condition is achieved

Artificial Immune System and Particle Swarm Optimization Hybrid

AIS have the advantage to prevent the population from being trapped into local optimum. Besides, PSO has the ability to improve itself but tend to converge prematurely [6]. Therefore, the combination between AIS and PSO (AIS-PSO) is expected to improve the global search ability and avoid being trapped in local minima even though the population size is relatively small. Hence, The AIS-PSO pseudocode is described in the following steps.

Step	Process
1	Select the best particles from PSO to be half of AIS initial population, N_1
2	Generate randomly other half of initial population of Abs, N_2
3	Combine N_1 and N_2 and compute fittest values of each Ab
4	Generate clones by cloning all cells in Ab population
5	Mutate the clone population to produce a mature clone population
6	Evaluate the affinity value for each clone in the population
7	Select the best Ab to compose the new Ab population
8	Repeat steps 4 to 7 until pre-defined stopping condition is achieved

Half Best Insertion Artificial Immune System

In AIS, clonal selection adapt B-cells (and T-cells) to kill the invader through affinity maturation by hypermutation. However, the adaptation requires B-cells to be cloned many times [7, 8], and the hypermutation process cannot always guarantee that the next generation will provide better solution. The stochastic factor (randomization) at times can even produce worse result from previous solution. Therefore, N number of the best Abs from the previous generation can be combined with the initial random Abs of the next generation to compose a new population for that next generation. This method known as Half Best Insertion (HBI) is expected to improve the convergence of the CSA algorithm. The HBI algorithm is described in the following steps.

Step	Process
1	Generate an initial random population of antibodies, Abs
2	Compute the fittest value of each Ab according to fitness function
3	Generate clones by cloning all cells in the Ab population
4	Mutate the clone population to produce a mature clone population
5	Evaluate the affinity value for each clone in the population and select N number of best Abs, α
6	Generate next generation of initial random Abs and include α
7	Repeat steps 2 to 6 until pre-defined stopping condition is achieved

Single Best Remainder Artificial Immune System

Hypermutation of good Abs in HBI algorithm would tend to produce bad solution. Thus, the Single Best Remainder (SBR) algorithm tries to avoid hypermutation process on the selected good Abs that produce worse solution due to stochastic factor. Therefore, the best Abs from previous generation is kept in global memory as single best antibody which is not affected by the next affinity maturation and hypermutation processes. The global single best antibody will be updated through generation and used in the next generation if the hypermutation result converges prematurely in the search space. Therefore, SBR is proposed in order to improve the convergence and accuracy of the CSA algorithm. The SBR algorithm is described in the following steps.

Step	Process
1	Generate an initial random population of Abs
2	Compute the fittest value of each Ab according to fitness function
3	Generate clones by cloning all cells in the Ab population
4	Mutate the clone population to produce a mature clone population
5	Evaluate the affinity value for each clone in the population
6	Select the best Ab, A_m, in 5 as global memory and repeat steps 1 to 5
7	Repeat steps 1 to 5 and compare the best Ab obtained with A_m
8	The best Ab from 7 is updated as the global memory, A_m
9	Repeat steps 1 to 9 until pre-defined stopping condition is achieved

All methods described above are evaluated using three mathematical test functions. The termination criteria for all methods will be met if minimum error value is achieved or maximum number of evaluation allowed is exceeded.

III. Experiments on Test Function

The computing platform used for the experiment is AMD Phenom 9600B Quad-Core CPU running at 2.30 GHz, 2GB of RAM and Windows Vista Enterprise operating system. Each algorithm is evaluated based on 200 iterations. The minimum error is set as 1e-25. The population size P_0 is set to 20.

In the HBI, antibodies and memory size of 50% are maintained. At first iteration, CSA is used to obtain the first solution. Then, for the next iteration, half of the population is composed by the half best antibodies after hypermutation and the other half is given by randomized Abs. The new population then goes through the affinity maturation process similar to CSA.

Then, in SBR, similar to HBI, CSA is used to obtain the first solution. Then, for the next iteration, one best antibody, A_m is kept as global memory. This A_m will never go through affinity maturation process, but will be assigned as a reference (memory) in case the hypermutation process produces worse solution.

The algorithm is evaluated using three benchmark functions (objective functions) which are described as follows.

Rastrigin's Function:

Rastrigin's function is mathematically defined as follows.

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

where $-5.12 \le x_i \le 5.12$, $i = 1 \ldots, n$

and global minimum is located at the origin and its function value is zero.

De Jong's Function:

De Jong's function is mathematically defined as follows.

$$f_2(x) = \sum_{i=1}^{n} x_i^2$$

where $-5.12 \le x_i \le 5.12$, $i = 1 \ldots, n$

and global minimum is located at the origin and its function value is zero.

Griewangk's Function:

$$f_3(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

where $-600 \le x_i \le -600$, $i = 1 \ldots, n$

and global minimum is located at the origin and its function value is zero.

IV. Result and Discussion

The results for the test functions are shown in Fig. 2 to 4 and Table I. For Rastrigin's function, Fig. 2 shows PSO suffers from premature convergence while GA is less accurate in giving the fitness value. On the other hand, SBR gives good fitness value while PSO-AIS and HBI is almost comparable in performance although the rate of convergence of PSO-AIS is very low. Alternatively, the performance of CSA is moderate with the fitness value at 1.56E-10.

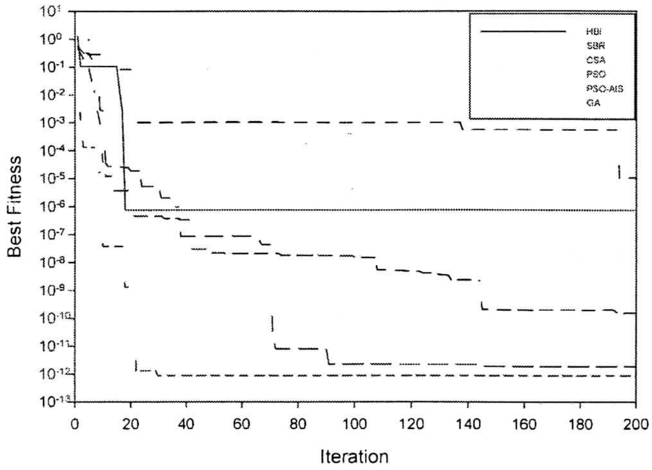

Figure 2. Algorithms evaluation on Rastrigin's function

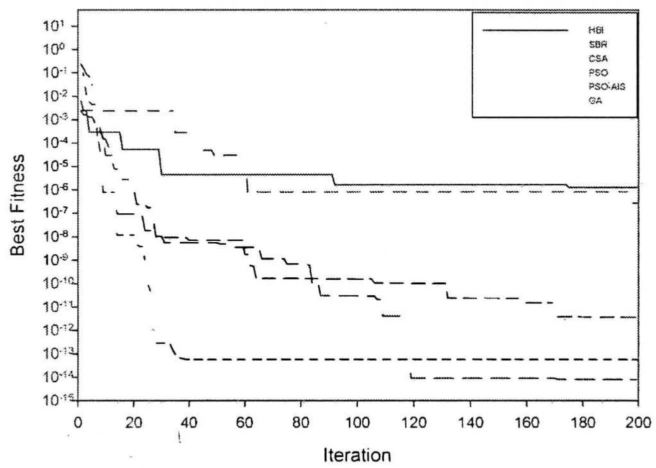

Figure 3. Algorithms evaluation on De Jong's function

De Jong's test function simulation results in Fig. 3 prove that SBR is superior in achieving the lowest solution of 8.00E-15 and followed by PSO and CSA. Again, PSO is clearly affected by premature convergence. HBI, PSO-AIS and GA attained almost comparable results.

Figure 4 also shows that PSO converges rapidly to achieve the lowest best fitness value among all algorithms. The second best solution is given by SBR which is comparable to the result obtained from CSA. PSO-AIS gave the least desirable result followed by GA.

The best fitness value obtained from each algorithm is shown in Table I with the respective test functions; TF1, TF2 and TF3.

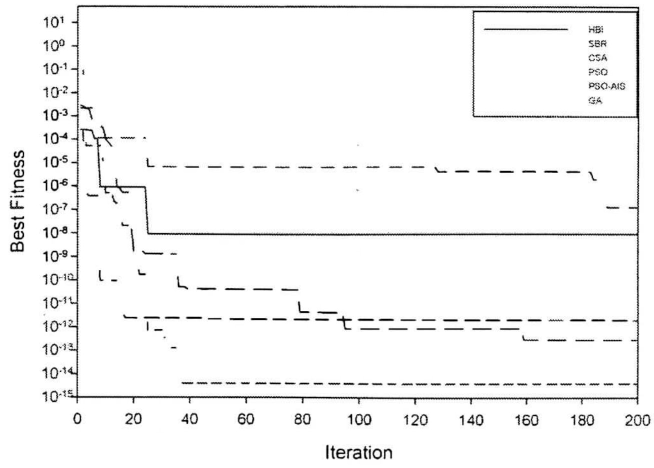

Figure 4. Algorithms evaluation on Griewangk's function

TABLE I. RESULT OF MINIMUM SOLUTION

TF	HBI	SBR	CSA	PSO	PSO-AIS	GA
1	7.57E-07	1.86E-12	1.56E-10	8.76E-13	2.89E-06	1.11E-05
2	1.23E-06	8.00E-15	3.69E-12	5.70E-14	9.65E-06	2.64E-07
3	9.66E-09	2.87E-13	1.97E-12	4.00E-15	1.05E-04	1.36E-07

V. CONCLUSION

In this paper, we proposed a memory-based clonal selection AIS strategy using local memory for each iteration, known as SBR and HBI. While PSO is fast in obtaining the fitness value, it suffers from premature convergence. Thus, simulation results clearly showed that the best result is given by SBR.

ACKNOWLEDGMENT

The authors would like to thank The Ministry of Higher Education Malaysia for the financial support under Fundamental Research Grant Scheme No. FRGS/2010/FKEKK/TK03/1 – F0085.

REFERENCES

[1] J. Kennedy and R. Eberhart, 1995. Particle Swarm Optimization. Proc. IEEE Int. Conf. Neural Networks, 4:1942-1948.

[2] J. Kennedy and R. Eberhart, 1995. Swarm Intelligence. 1st Edition. , Academin Press., San diego. CA.

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

[4] L. N. De Castro and J. Timmis, 2002. Artificial Immune System: A New Computational Approach. Springer-Verlag, NewYork.

[5] A. Acan and A. Unveren, "A Memory-Based Colonization Scheme for Particle Swarm Optimization," IEEE Congress on Evolutionary Computation, pp. 1965-1972, 2005.

[6] J. Vesterstrom and R. Thomsen, "A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems," pp. 1980-1987.

[7] S. M. Garret, "How Do We Evaluate Artificial Immune System", Evoluationary Computation, vol. 13, pp. 145-178.

[8] F. M. Burnet, 1969. The Clonal Selection Theory of Acquired Immunity. Cambridge University Press.