Hindawi Publishing Corporation
Applied Computational Intelligence and Soft Computing
Volume 2011, Article ID 210918, 8 pages
doi:10.1155/2011/210918

Research Article

Design of Fixed and Ladder Mutation Factor-Based Clonal Selection Algorithm for Solving Unimodal and Multimodal Functions

Suresh Chittineni,[1] A. N. S. Pradeep,[1] Dinesh Godavarthi,[1] Suresh Chandra Satapathy,[1] S. Mohan Krishna,[2] and P. V. G. D. Prasad Reddy[3]

[1] *Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, Andhra Pradesh, India*
[2] *Gitam University, Visakhapatnam, India*
[3] *Andhra University Engineering College, Visakhapatnam, Andhra Pradesh, India*

Correspondence should be addressed to Suresh Chandra Satapathy, sureshsatapathy@gmail.com

Received 6 May 2011; Accepted 14 July 2011

Academic Editor: Maoguo Gong

Clonal selection algorithms (CSAs) is a special class of immune algorithms (IA), inspired by the clonal selection principle of the human immune system. To improve the algorithm's ability to perform better, this CSA has been modified by implementing two new concepts called fixed mutation factor and ladder mutation factor. Fixed mutation factor maintains a constant factor throughout the process, where as ladder mutation factor changes adaptively based on the affinity of antibodies. This paper compared the conventional CLONALG, with the two proposed approaches and tested on several standard benchmark functions. Experimental results empirically show that the proposed methods ladder mutation-based clonal selection algorithm (LMCSA) and fixed mutation clonal selection algorithm (FMCSA) significantly outperform the existing CLONALG method in terms of quality of the solution, convergence speed, and solution stability.

1. Introduction

Artificial Immune System (AIS) is one of the bioinspired approaches for solving the real complex and difficult optimization problems. The AIS is greatly reinforced by the human immune system. In humans, the immune system is responsible for protection from pathogens. De Castro and von Zuben proposed a clonal selection algorithm (CSA) based on the clonal selection principle and the affinity maturation process [1]. CLONALG (clonal algorithm) is an artificial immune algorithm [2] based on clonal selection principle. CLONALG is used to optimize functions [1]. CLONALG has global searching ability as it uses the principle of clonal expansion and affinity maturation as the main forces of the evolutionary process.

In this paper, we present an improved version of the immune system model based on the clonal selection theory. Two algorithms LMCSA (ladder mutation factor-based clonal selection algorithm) and FMCSA (fixed mutation factor-based clonal selection algorithm) are proposed by introducing two novel immune mutation factors and are applied to unimodal and multimodal functions optimizations. The results illustrate that the proposed algorithms have a remarkable performance over basic CLONALG.

The remainder of this paper is organized as follows. Section 2 briefly discusses the basic steps in clonal selection optimization algorithm (CLONALG) and antibody diversity maintaining principles. Section 3 describes our modified versions of immune optimization algorithm with the introduction of mutation factor and its details. Section 4 gives further explanations, experimental analysis, simulation results to several benchmark problems, and comparisons of our proposed algorithms with the conventional CLONALG. Section 5 concludes our paper with some remarks and conclusions.

2. Basic Immune Optimization Algorithm (CLONALG)

CLONALG is a population-based metaheuristic algorithm whose search power relies on its mutation operator. In our proposed work the main thrust is given to these mutation operators while developing better algorithms. The clonal selection algorithm (CSA) reproduces individuals with high affinities and selects improved maturate progenies. This strategy suggests that this algorithm performs a greedy search, where single members will be locally optimized, and the newcomers yield a broader exploration of the search space. This characteristic makes the CSA very suitable for solving optimization tasks. The basic steps and working of immune optimization algorithm (CLONALG) is described as follows.

Step 1 (antibody pool (AB) initialization). Initially, an Antibody Pool (AB) is created with N antibodies chosen randomly in the search space. Antibodies are represented by the variables of the problem $(\mathrm{ab}_1, \mathrm{ab}_2, \ldots, \mathrm{ab}_N)$ which are potential solutions to the problem.

Step 2 (selection). For each antibody (ab_i), its corresponding affinity is determined. These antibodies are then sorted according to the affinity calculated. And n antibodies are selected having the highest affinity.

Step 3 (cloning). Cloning is one of the key aspects in AIS. It is the process of producing similar populations of genetically identical individuals. The best selected n antibodies will be replicated in proportionate to their antigenic affinity. The replicated antibodies, that is, clones, are maintained as a separate clone population C. The number of clones for each antibody can be calculated by the following equation:

$$N_c = \sum_{i=1}^{n} \mathrm{round}\left(\frac{\beta \cdot N}{i}\right), \tag{1}$$

where N_c is the total number of clones [3] generated, β is a multiplying factor, N is the size of Antibody Pool (AB), and round $(\cdot)$ is the operator that rounds its argument towards the closest integer. Clone size of each selected antibody is represented by each term of this sum. The higher the affinity, the higher the number of clones generated for the selected antibody [4].

Step 4 (affinity mutation). The clone population C is now subjected to mutation process which is inversely proportional to its antigenic affinity measurement function. This mutation helps for low-affinity antibodies to mutate more in order to improve its affinity. The mutations always result in better-affinity antibodies. For Gray's coding, uniform mutation, the Gaussian mutation or Cauchy's mutation can be used. In this Paper, self-adaptive mutation using the Gaussian distribution is used for making a search in the area surrounding the cell with high probability. And it has an outstanding ability of both local and global searching. The Gaussian mutation operator [2] can be described as follows:

$$\begin{aligned} \theta_i^j &= \theta_i^j \times \exp\left(\tau_1 \times N(0,1) + \tau_2 \times N_j(0,1)\right), \\ \mathrm{ab}_i^j &= \mathrm{ab}_i^j + \theta_i^j \times N_j(0,1), \\ \tau_1 &= \left(\sqrt{2 \times \sqrt{D}}\right)^{-1}, \qquad \tau_2 = \left(\sqrt{2 \times D}\right)^{-1}, \end{aligned} \tag{2}$$

where $\theta_i = \{\theta_1, \theta_2, \ldots, \theta_D\}$, $i = 1, 2, 3, \ldots, N_c$, $j = 1, 2, \ldots, D$. The parameter θ_i^j is the mutation step of antibody ab_i^j, τ_1 and τ_2 are the whole step and the individual step, respectively.

Then, the affinities of the mutated clones are calculated. The better-affinity mutations are stimulated while the worse are restrained when antibody undergoes affinity mutation. The higher affinity values are taken for next generation while the Lower-affinity antibodies are deleted.

Step 5 (antibody diversity maintenance). Inspired by the vertebrate immune system mechanism called antibodies restraint, the processes of suppression and supplementation are defined in CLONALG. This step maintains diversity and helps to find new solutions that correspond to new search regions by eliminating some percentage of the worst antibodies in the population and replacing with the randomly generated new antibodies. This helps the algorithm to avoid being trapped to local optimal solutions. In antibodies restraint [5], for every iteration, the similar antibodies are removed, and randomly generated antibodies are introduced in the place of removed antibody of the Antibody Pool (AB).

The pseudocode of antibody restraint's redundancy removal is explained below for better clarity.

Step 1. For every antibody, affinity is computed and then sorted based on affinity values in descending order. Let an antibody vector be $\mathrm{Pop}(\mathrm{ab}_1, \mathrm{ab}_2, \ldots, \mathrm{ab}_N)$; N is the total number of antibody in generation.

Step 2. Set $i = 1$, $j = N$
Repeat:
Checking: Is AB(i) and AB(N) Similar or not?
If AB(i) is similar with AB(N)
Delete AB(N) in population
Else
$j = j - 1$
End if
$i = i + 1$
Until $i = N - 1$.

Step 3. If number of the antibodies $< N$

Add new antibodies to AB.

Step 6 (convergence check). Repeat Steps 2 to 5, until the following conditions are met.

Table 1: Benchmark test functions of dimension D. Test functions each of them has a global minimum value of 0.

	Test function	Domain range
$f1$	$\sum_{T=1}^{D} x_1^2$	$[-100\text{–}100]^D$
$f2$	$\sum_{T=1}^{D} \lvert x_1\rvert + \prod_{t=1}^{D} \lvert x_1\rvert$	$[-10\text{–}10]^D$
$f3$	$\sum_{T=1}^{D} (\sum_{j}^{t} x_j)^2$	$[-100\text{–}100]^D$
$f4$	$10^4 x_1^2 + \sum_{t=2}^{D} x_1^2$	$[-100\text{–}100]^D$
$f5$	$\sum_{t=1}^{D-1} [100(x_{i+1} - x_1^2)^2 + (x_1 - 1)^2]$	$[-30\text{–}30]^D$
$f6$	$\sum_{t=1}^{D} [x_i + 0.5]^2$	$[-100\text{–}100]^D$
$f7$	$\sum_{t=1}^{D} i x_t^4 + \text{rand}[0.1)$	$[-1.28\text{–}1.28]^D$
$f8$	$\sum_{t=1}^{D} [x_1^2 - 10\cos(2\pi x_1) + 10]$	$[-5.12\text{–}5.12]^D$
$f9$	$-20\exp((-0.2\sqrt{(1/D)\sum_{t=1}^{D} x_1^2}) - \exp((1/D)\sum_{t=1}^{D}\cos(2\pi x_1) + 20 + e$	$[-32\text{–}32]^D$
$f10$	$\Sigma_{t=1}^{D}$ $[-x_i \sin\sqrt{\lvert x_1\rvert}+$ D.418.98288727433]	$[-500\text{–}500]^D$
$f11$	$(1/4000)\sum_{t=1}^{D} x_1^2 - \prod_{t=1}^{D}\cos(x_i/\sqrt{t}) + 1$	$[-600\text{–}600]^D$
$f12$	$\sum_{t=1}^{N} x_t^2 + (\sum_{t=1}^{N} 0.5t \cdot x_1^2) + (\sum_{t=1}^{N} 0.5t \cdot x_1^4)$	$[-10\text{–}10]^D$

(i) Algorithm undergoes a specified number of iterations.

(ii) The optimal solutions in memory cells are not improved in given generations.

3. Proposed Fixed Mutation Factor- and Ladder Mutation Factor-Based Clonal Selection Algorithm (FMCSA and LMCSA)

In the basic CLONALG, the initially best and worst antibodies are identified; the process of cloning is applied to both the best and the worst antibodies such that cloning rate is high to the best antibodies and less to the worst antibodies [4]. Therefore, more clones are produced for the antibody that has the highest affinity. Further, the worst antibodies are mutated in order to make them better. By doing this, the worst antibodies can improve; however, no care is taken for the best antibodies. Since more population of the best antibodies also exist in that pool, there is a chance of faster convergence if best antibodies are also taken care properly. Otherwise, these can lead to local optima and the low convergence rate, resulting in poor performance of the algorithm.

In this paper, two novel methods are introduced to solve this problem by properly nurturing the best antibodies. The basic flowchart for these methods is given in Figure 1.

(A) Fixed Mutation Factor-Based Clonal Selection Algorithm (FMCSA). Like in CLONALG, the best antibody is cloned according to the cloning rate (β), and clones of best antibodies are generated. In this process, mutation is done to some of the best antibodies also along with the worst antibodies. A few percentages of best antibodies that are cloned are taken and are mutated along with the worst antibodies. So, a fixed mutation factor (γ) is defined as the percentage of the best cloned antibodies in clone population (C) that are to be considered for mutation. This mutation factor (γ) is fixed throughout the process. The number of the best antibodies to be considered for mutation in each antibody's clone population (CMUT_{AB}) is calculated as follows:

$$\text{CMUT}_{\text{AB}} = \text{Ceil}(\gamma * C_{\text{AB}}), \tag{3}$$

where CMUT_{AB} is the number of the best antibodies to be considered for mutation in each antibody's clone population. γ is the fixed mutation factor, C_{AB} is the total number of antibodies in each clone population of an antibody, and $\text{Ceil}(\cdot)$ is the operator that rounds its argument towards the nearest integers.

For example, for 5 initial antibodies after performing Steps from 1 to 3, let the clone population be (5,4,3,2,1). Let the fixed mutation factor γ be 0.3. So, upon using (3), $\text{CMUT}_{\text{AB}} = 2$. Hence, 2 best antibodies are considered out of 5 initial cloned antibodies for mutation. As the worst antibodies are having less antigenic affinity, they are cloned less in number as in basic CLONALG [1]. In general, for FMCSA algorithm, the fixed mutation factor (γ) is chosen to be very small.

(B) Ladder Mutation Factor-Based Clonal Selection Algorithm (LMCSA). In the fixed mutation factor approach, the mutation factor remains constant. When the affinity difference between the worst antibody and the best antibody is high, then the worst antibodies try to go and follow the best as in any other evolutionary strategies. But, when their affinity difference is very less, all the worst and the best antibodies are in the same surrounding area of the search space, or, in other words, the worst antibody has come closer to the best antibody's area [8]. At this time, the further improvement may not achieve at faster rate. This can be enhanced by considering few additional antibodies for mutation. A chance of better convergence can be attained by adaptively incrementing the percentage of best cloned antibodies for mutation as explained in Figure 2. The mutation factor can be incremented adaptively based on the affinity measure. This mutation factor is proportional to the affinity of the antibodies; that is, if the ratio of the affinities between the best and worst antibody is less than the threshold value (μ), then mutation factor should be incremented.

The following pseudocode is included in Step 4 of the basic CLONALG:

If

$$\begin{gathered}(\text{aff}[\text{ab}_b]/\text{aff}[\text{ab}_w]) < \mu \\ \gamma = \gamma \times \rho\end{gathered} \tag{4}$$

ρ is a parameter, changed dynamically as follows:

$$\rho = \left(\frac{\text{iter}}{\text{maxiter}}\right) \times \alpha \tag{5}$$

End

where μ is the threshold value. ab_b is the best antibody in the Antibody Pool. ab_w is the worst antibody in the Antibody

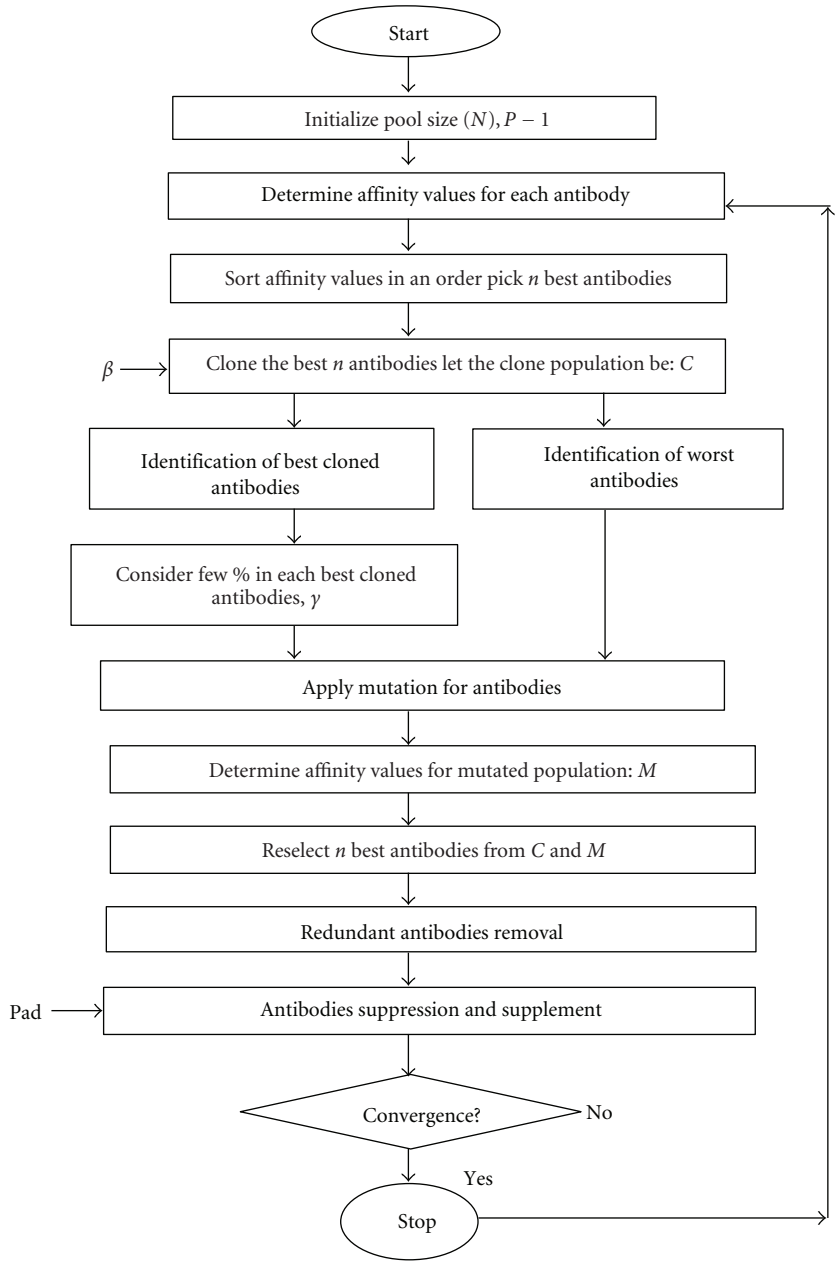

Figure 1: Basic Flow chart for FMCSA and LMCSA.

Pool. αt5 α constant multiplier depends on the problem type. γ is the mutation factor. iter stands for current iteration. maxiter stands for maximum number of iterations. aff$(\cdot)$ is a function used for calculating the affinity of the antibody.

4. Benchmark Test Functions: Results and Analysis

(A) Benchmark Functions for Simulation. A suite of twelve standard and well-known benchmark functions [6, 7, 9, 10] are taken into consideration to test the effectiveness and the efficiencies of the proposed approaches FMCSA and LMCSA with the basic CLONALG. All these benchmark functions are meant to be minimized. The mentioned benchmark test functions cover different kinds of optimization problems. They can be divided into two categories. The first one is the category of the unimodal function, which is a symmetric model with a single minimum; in Table 1, $f1$–$f7$ are unimodal functions. The functions $f8$–$f12$ in Table 1 belong to the category of multimodal functions with many local minima.

(B) Experimental Setup. The approaches that are described earlier have been coded using the MATLAB Scripting

Table 2: Statistical means and standard deviations of the solutions of twelve Benchmark test functions, listed in Table 1 for the CLONALG, FMCSA, and the LMCSA.

Functions	Expressions	Basic CLONALG Mean ± StD	FMCSA Mean ± StD	LMCSA Mean ± StD
Sphere	$f1$	$1.0463e-002 \pm 1.023e-004$	$1.103e-003 \pm 1.836e-005$	$7.804e-005 \pm 1.8025e-007$
Schwefel's Problem 2.22	$f2$	2.3359 ± 1.45602	$6945e-001 \pm 4.5602e-001$	$2.359e-001 \pm 1.9721-002$
Schwefel's Problem 1.2	$f3$	2.277 ± 3.49	$7.514e-001 \pm 1.61$	$9.1e-002 \pm 7.11e-001$
Tablet	$f4$	$6.85e+002 \pm 13.1e+002$	$3.17e+001 \pm 9.7e+001$	1.089 ± 3.58
Generalized Rosenbrock	$f5$	$2.409e+003 \pm 1.31e+003$	$1.17e+002 \pm 9.25e+001$	$8.9 \pm 1.14e+001$
Step	$f6$	$2.217e-001 \pm 2.99e-001$	$8.709e-002 \pm 2.46e-004$	$2.076e-002 \pm 3.40e-003$
Quartic Noisy	$f7$	$1.762e-001 \pm 1.07e-001$	$4.031e-002 \pm 3.71e-002$	$3.408e-004 \pm 4.39e-003$
Generalized Rastrigin	$f8$	3.89757 ± 6.201	2.4420 ± 3.15	$2.390e-001 \pm 2.1604$
Ackley	$f9$	2.6739 ± 1.1735	$1.5230e-003 \pm 2.78e-002$	$9.7821e-003 \pm 3.657e-003$
Generalized Schwefel's Problem 2.26	$f10$	$3.66e+002 \pm 9.61e+001$	$1.8e+001 \pm 3.1e+001$	$9.12e-001 \pm 1.21e-001$
Griewank	$f11$	$1.178e-004 \pm 3.4e-008$	$6.159e-007 \pm 2.3e-010$	$1.3912e-011 \pm 2.4e-012$
Zakharov	$f12$	$3.72e+001 \pm 6.9975$	$1.23e-001 \pm 2.71$	$9.141e-04 \pm 1.21e-002$

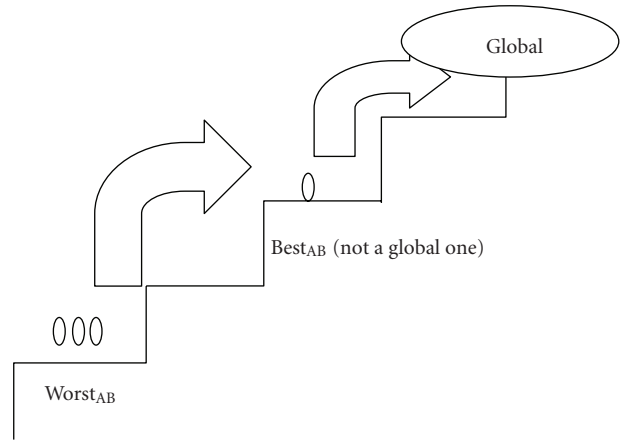

Figure 2: Schematic diagram of the ladder mutation factor concept. Worst$_{AB}$ represents the worst antibodies. Best$_{AB}$ represents the best antibody achieved so far. 0 represents an antibody.

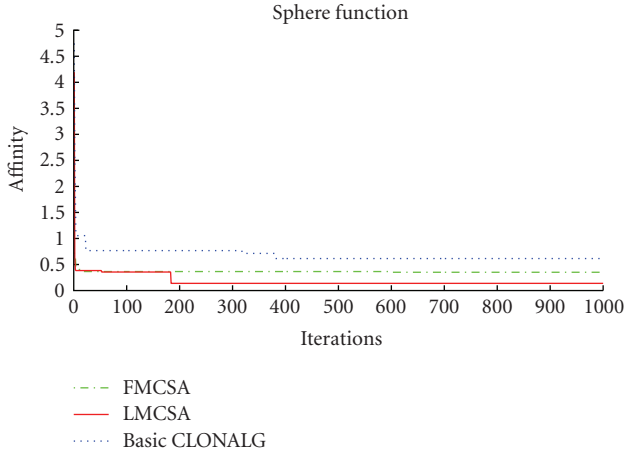

Figure 3: Convergent performances on sphere function ($f1$).

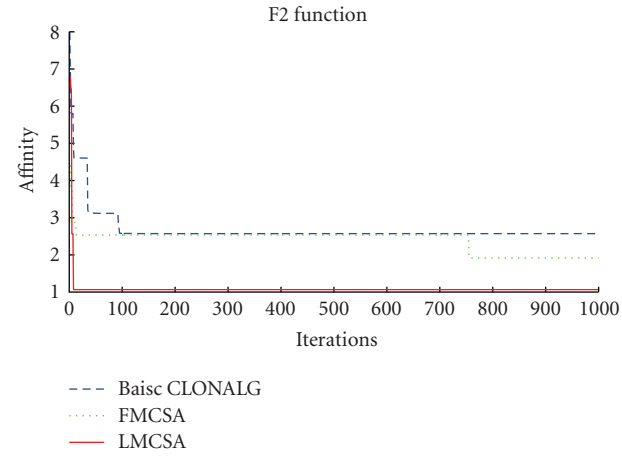

Figure 4: Convergent performances on Schwefel's Problem 2.22 function ($f2$).

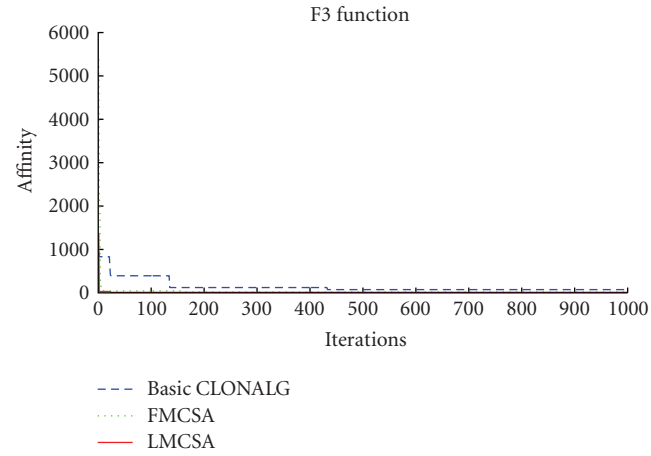

Figure 5: Convergent performances on Schwefel's Problem 1.2 function ($f3$).

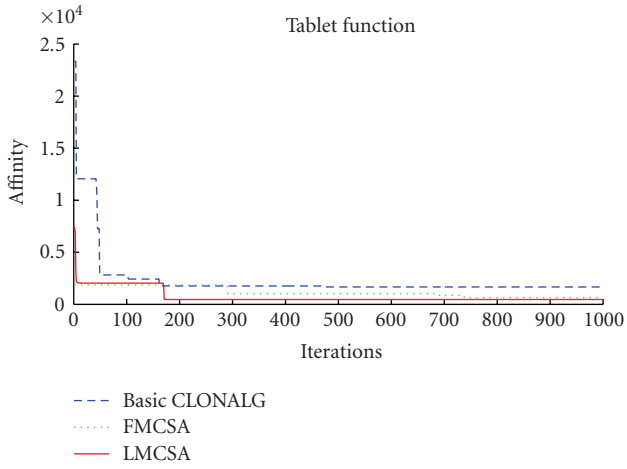

Figure 6: Convergent performances on tablet function ($f4$).

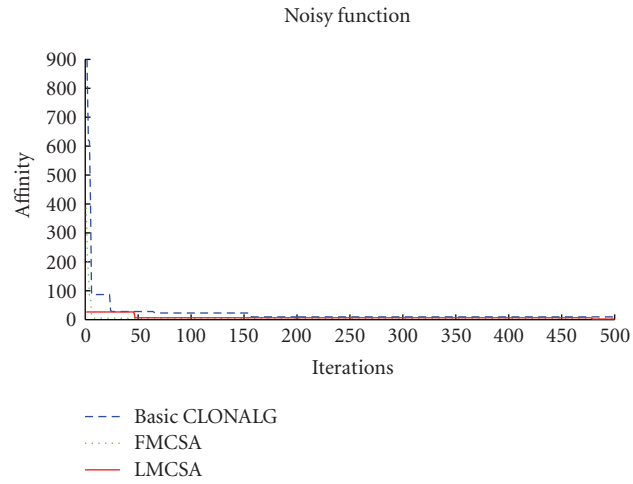

Figure 9: Convergent performances on the noisy function ($f7$).

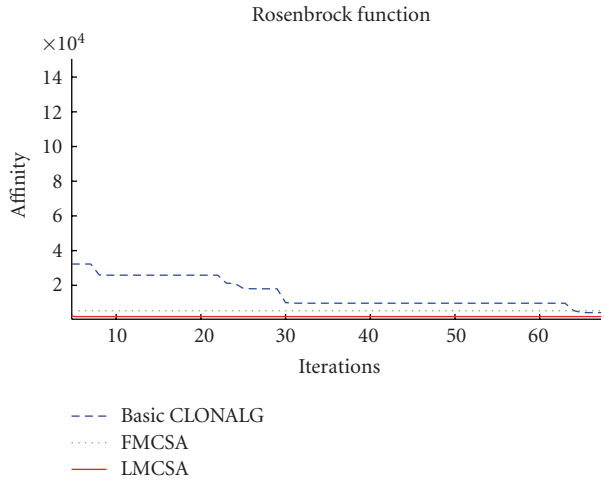

Figure 7: Convergent performances on Rosenbrock's Function ($f5$).

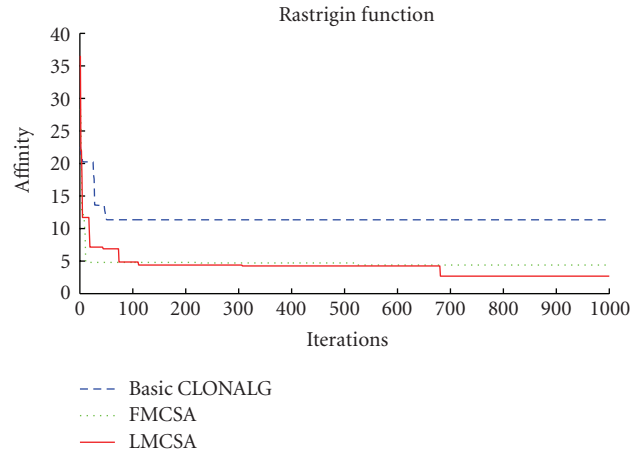

Figure 10: Convergent performances on Rastrigin's function ($f8$).

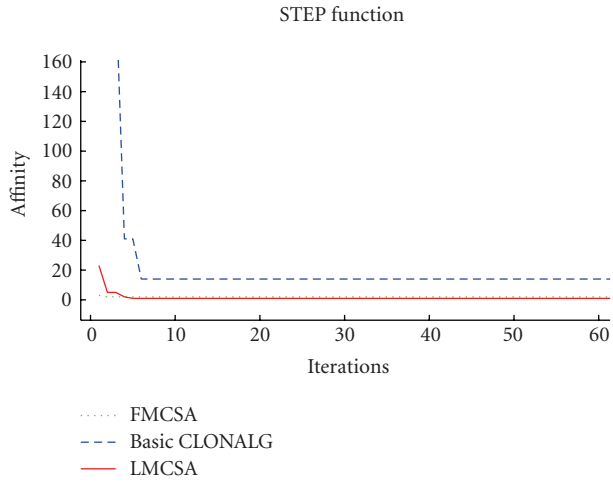

Figure 8: Convergent performances step function ($f6$).

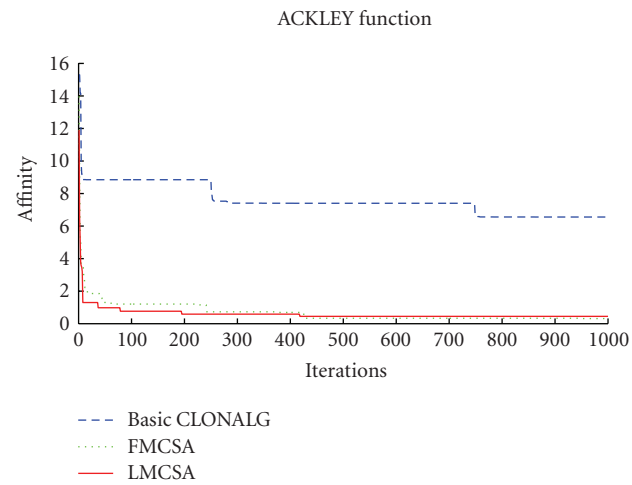

Figure 11: Convergent performances on Ackley's function ($f9$).

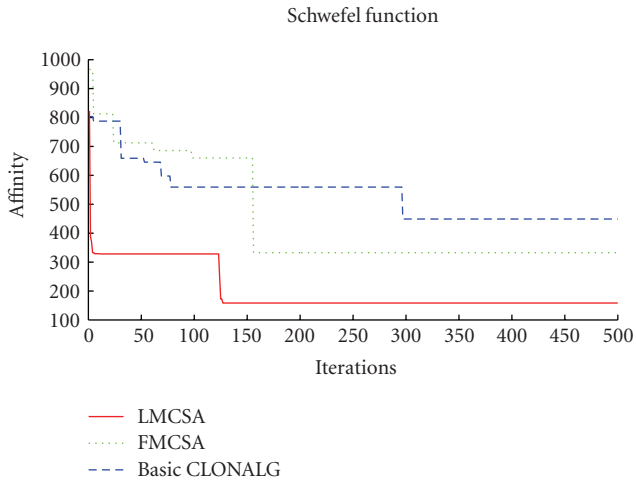

Figure 12: Convergent performances on generalized Schwefel's Problem 2.26 ($f10$).

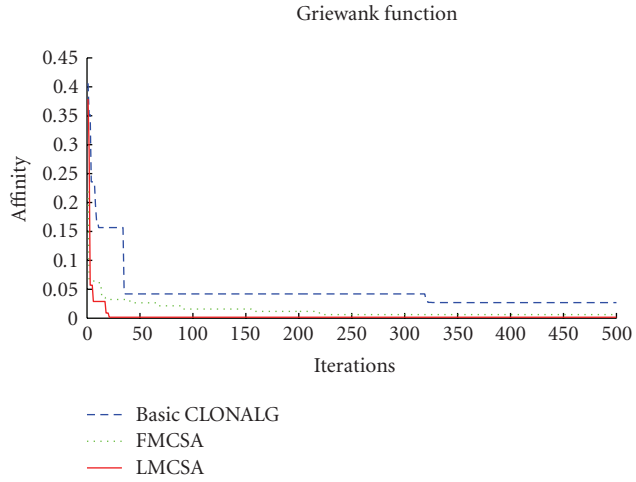

Figure 13: Convergent performances on Griewank's function ($f11$).

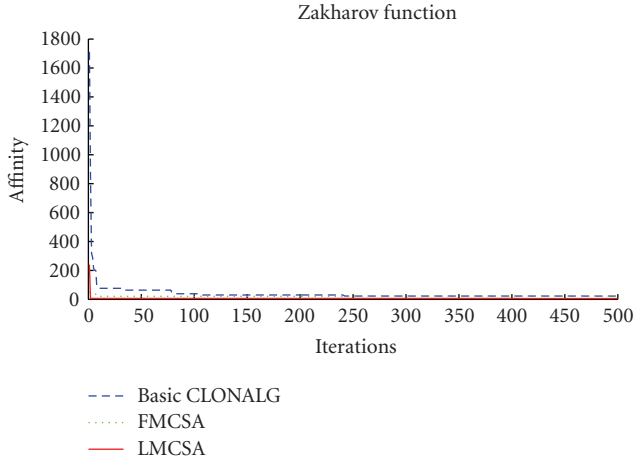

Figure 14: Convergent performances on Zakharov's function ($f12$).

language, and all experiments took place on a 1.8 GHz Intel Core 2 Duo processor, 2 GB of RAM, and on Windows XP operating system. Each algorithm is evaluated for 1000 iterations as the termination criteria.

The following simulation conditions are used.

(i) Initial population or Antibody Pool Size, AB = 50.

(ii) Clone multiplying factor's range $\beta = [0.5–1]$.

(iii) Type of mutation used: Gaussian.

(iv) Gaussian mutation probability $P_{\text{mg}} = 0.01$.

(v) Percentage of suppression $P_{\text{sup}} = 0.2$.

(vi) The affinity threshold $\sigma = 10^{-3}$.

(vii) Number of iterations = 1000.

(viii) Fixed mutation factor (γ) in FMCSA = 0.20.

(ix) μ and α varies from the problem's domain range.

(x) Number of dimensions taken for each Benchmark function = 10.

(C) Benchmark Results and Analysis. In this section, the results for the 12 benchmark test functions are given to show the merits of the proposed FMCSA and LMCSA. The experimental results in terms of the mean affinity value, the best cost value, and the standard deviation are summarized in Table 2, and Figures 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, and 14 give the performance curves for each function. The analyses of results are done category-wise as presented below.

Category 1 (unimodal functions). The main purpose of testing these functions from $f1$ to $f7$ is to measure the convergence rate of searching. For these functions, the results in terms of the mean affinity values of the LMCSA are much better than that of FMCSA and the conventional CLONALG. However, FMCSA outperforms CLONALG. The lower standard deviation values in LMCSA show the stability of the algorithm. It can be verified from performance curves given in Figures 3 to 9 that LMCSA has a very fast rate of convergence to solution compared to FMCSA and CLONALG algorithms. For the function $f6$, both FMCSA and LMCSA performed similarly.

Category 2 (multimodal functions with many local minima). For these kinds of functions, more chances are there for being easily trapped at some local minimum. Functions $f8–f12$ are multimodal functions with many local minima. The experimental results for these functions are tabulated in Table 2. It can be clearly seen from Table 2 and from Figures 10 to 14 that the results in terms of the mean affinity values, standard deviation, and convergence rate of the LMCSA are much better than those of the proposed method FMCSA and the conventional CLONALG. And the results obtained from FMCSA are better than CLONALG in all cases. For the function $f9$, both FMCSA and LMCSA perform similarly.

5. Conclusions

In this paper, we proposed two novel approaches, fixed mutation clonal selection algorithm (FMCSA) and ladder mutation clonal selection algorithm (LMCSA). Our objective is to increase the searching area by increasing a few numbers of antibodies that undergo mutation so as to further improve the performance of basic CLONALG. On solving a suite of benchmark functions, FMCSA performs better than CLONALG, and LMCSA outperforms both FMCSA and CLONALG. Simulation results on standard benchmark problems have shown that the proposed methods are useful techniques to solve complex optimization problems.

References

[1] L. N. de Castro and F. J. von Zuben, "Learning and optimization using the clonal selection principle," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 3, pp. 239–251, 2002.

[2] X. Xuesong and Z. Jing, "An improved immune evolutionary algorithm for multimodal function optimization," in *Proceedings of the 3rd International Conference on Natural Computation (ICNC '07)*, pp. 641–646, August 2007.

[3] N. Cruz-Cortés, D. Trejo-Pérez, and C. A. Coello Coello, "Handling constraints in global optimization using an artificial immune system," in *Proceedings of the 4th International Conference on Artificial Immune Systems (ICARIS '05)*, pp. 234–247, August 2005.

[4] L. N. de Castro and J. Timmis, *An Introduction to Artificial Immune Systems: A New Computational Intelligence Paradigm*, Springer, 2002.

[5] L. Pan and Z. Fu, "A clonal selection algorithm for open vehicle routing problem," in *Proceedings of the 3rd International Conference on Genetic and Evolutionary Computing (WGEC '09)*, pp. 786–790, October 2009.

[6] S. H. Ling, H. H. C. Iu, K. Y. Chan, H. K. Lam, B. C. W. Yeung, and F. H. Leung, "Hybrid particle swarm optimization with wavelet mutation and its industrial applications," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 38, no. 3, pp. 743–763, 2008.

[7] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[8] J. Timmis, C. Edmonds, and J. Kelsey, "Assessing the performance of two immune inspired algorithms and a hybrid genetic algorithm for function optimisation," in *Proceedings of the 2004 Congress on Evolutionary Computation (CEC '04)*, pp. 1044–1051, June 2004.

[9] K. A. Al-Sheshtawi, H. M. Abdul-Kader, and N. A. Ismail, "Artificial immune clonal selection algorithms: a comparative study of CLONALG, opt-IA, and BCA with numerical optimization problems," *International Journal of Computer Science and Network Security*, vol. 10, no. 4, pp. 24–30, 2010.

[10] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.