Proceedings of IEEE Congress on Evolutionary Computation (CEC'02), vol. **1**, pp. 699-674, 2002, May, Hawaii.
– **DRAFT** –

# An Artificial Immune Network for Multimodal Function Optimization

Leandro N. de Castro & Jon Timmis
L.N.deCastro@ukc.ac.uk, J.Timmis@ukc.ac.uk
Computing Laboratory, University of Kent at Canterbury (UKC),
Kent, Canterbury, UK, CT2 7NF, Phone: +44 (0) 1227 823636

**Abstract - This paper presents the adaptation of an immune network model, originally proposed to perform information compression and data clustering, to solve multimodal function optimization problems. The algorithm is described, theoretically and empirically compared with similar approaches from the literature. The main features of the algorithm are automatic determination of the population size, combination of local with global search (exploitation plus exploration of the fitness landscape), defined convergence criterion, and capability of locating and maintaining stable local optima solutions.**

## I. INTRODUCTION

In spite of the broad applicability of artificial immune systems (AIS) to innumerable domains, the field is only now, around 15 years after its "birth date", receiving a more careful attention from a theoretical and formal perspective. Y. Ishida and collaborators edited the first book in the year 1998 [1] on immune-based systems. This text was written in Japanese, what considerably restricted its diffusion. In early 1999, another volume [2] with a collection of papers on theoretical immunology and artificial immune systems was edited.

Although there are several authors publishing works in the field and defining novel algorithms as artificial immune systems, there is no consensus about what is an AIS, when it emerged, and most importantly, what characterizes an AIS and how to engineer one. A first textbook is now being proposed [3] as an attempt to answer to these questions and to aid novice and mature researchers in the field on the development and formalization of AIS. Basically, the authors argue that the shape-space approach [4] can be extended as a general framework to create abstract models of immune cells and molecules. In addition, several works from the literature, taken from various schools, were brought together in order to provide a set of general-purpose AIS algorithms. These include (but are not restricted to) a negative selection algorithm [5], a positive selection algorithm [6], a clonal selection algorithm [7], continuous immune network models [8], [9], and discrete immune network models [10], [11].

The model presented in this paper is based upon previous works from the literature and the formalism proposed in [3]. It is an adaptation of a discrete immune network algorithm originally developed to perform data analysis. The new version of the algorithm is evolutionary-like and has several interesting features: 1) population size dynamically adjustable, 2) exploitation and exploration of the search-space, 3) location of multiple optima, 4) capability of maintaining local optima solutions, and 5) defined stopping criterion.

## II. THE ORIGINS

The clonal selection and affinity maturation principles are used to explain how the immune system reacts to pathogens and how it improves its capability of recognizing and eliminating pathogens [12]. In a simple form, clonal selection states that when a pathogen invades the organism, a number of immune cells that recognize these pathogens will proliferate; some of them will become effector cells, while others will be maintained as memory cells. The effector cells secrete antibodies in large numbers, and the memory cells have long life spans so as to act faster and more effectively in future exposures to the same or a similar pathogen. During the cellular reproduction, the cells suffer somatic mutations with high rates and, together with a selective force, the higher affinity cells in relation to the invading pathogen differentiate into memory cells. This whole process of somatic mutation plus selection is known as affinity maturation.

To a reader familiar with evolutionary biology, these two processes of clonal selection and affinity maturation are much akin to the (macro-)evolution of species. There are a few basic differences however, between these immune processes and the evolution of species. Within the immune system, somatic cells reproduce in an asexual form (there is no crossover of genetic material during cell mitosis), the mutation suffered by an immune cell is proportional to its affinity with the selective pathogen (the higher the affinity, the smaller the mutation rate), and the number of progenies of each cell is also proportional to its affinity with the selective pathogen (the higher the affinity, the higher the number of progenies). Evolution in the immune system occurs within the organism and, thus it can be viewed as a micro-evolutionary process.

An immune algorithm, named CLONALG, was developed [7] to perform pattern recognition and optimization. The authors demonstrated empirically that this algorithm is capable of learning a set of input patterns by selecting, reproducing and mutating a set of "artificial immune cells". In [7] the authors showed the suitability of the algorithm for multimodal search and presented empirical results where it could outperform a fitness sharing strategy. All the steps involved in CLONALG are also seen in an evolutionary algorithm, allowing it to be characterized as an evolutionary algorithm inspired in the immune system. Note that there is an important conceptual difference between the clonal selection algorithm and an evolutionary algorithm. In the former, the theory of evolution is used to explain the behavior of the system, while in the latter it inspired its development.

In a subsequent work [10], it was proposed an AIS combining CLONALG with the immune network theory introduced in [13]. This model named aiNet has been successfully applied to several data compression and clustering applications [10], [14], including non-linear separable and high-dimensional problems. It also demonstrated to be a powerful strategy to be hybridized with neural networks in order to alleviate some neural network limitations such as model selection.

The same rationales that led to the development of CLONALG are motivations for the implementation of an optimization version of aiNet. First, it is possible to view clustering as an optimization problem where each cluster corresponds to a fitness peak of a subgroup of individuals within the whole population. In addition, aiNet is an extension of CLONALG with steps involving the interaction of the network cells with each other. The advantage of having steps in the algorithm that evaluate the degree of similarity among cells is that it is possible to maintain a dynamic control of the number of network cells, allowing the determination of more parsimonious solutions.

## III. AN OPTIMIZATION VERSION OF aiNet

In order to present an optimization version of aiNet (opt-aiNet) assume the following terminology:

- *Network cell*: individual of the population. In this case no encoding is performed, each cell is a real-valued vector in an Euclidean shape-space;
- *Fitness*: fitness of a cell in relation to an objective function to be optimized (either minimized or maximized). The value of the function when evaluated for the given cell;
- *Affinity*: Euclidean distance between two cells;
- *Clone*: offspring cells that are identical copies of their parent cell. The offspring will further suffer a somatic mutation so that they become variations of their parent.

The optimization version of aiNet (opt-aiNet) can be summarized as follows:

1. Randomly initialize a population of cells (the initial number of cells is not relevant)
2. **While** stopping criterion is not met **do**
2.1 Determine the fitness of each network cell and normalize the vector of fitnesses.
2.2 Generate a number *Nc* of clones for each network cell.
2.3 Mutate each clone proportionally to the fitness of its parent cell, but keep the parent cell. The mutation follows Eq. (1).
2.4 Determine the fitness of all individuals of the population.
2.5 For each clone, select the cell with highest fitness and calculate the average fitness of the selected population.
2.6 **If** the average error of the population is not significantly different from the previous iteration, then continue. **Else**, return to step 2.1
2.7 Determine the affinity of all cells in the network. Suppress all but the highest fitness of those cells whose affinities are less than the suppression threshold $\sigma_s$ and determine the number of network cells, named memory cells, after suppression.
2.8 Introduce a percentage *d*% of randomly generated cells and return to step 2.
3. **EndWhile**

The behavior of the new algorithm can be explained in a simple form. Steps 2.1 to 2.5: at each iteration, a population of cells is optimized locally through affinity proportional mutation (exploitation of the fitness landscape). The fact that no parent cell has a selective advantage over the others contributes to the multimodal search of the algorithm. Steps 2.6 to 2.8: when this population reaches a stable state (measured via the stabilization of its average fitness), the cells interact with each other in a network form, and some of the similar cells are eliminated to avoid redundancy. Also, a number of randomly generated cells is added to the current population (exploration of the fitness landscape) and the process of local optimization re-starts.

A number of interesting features of opt-aiNet can be outlined. 1) It presents a deterministic and elitist selection mechanism for each clone. 2) The cardinality of the population is automatically determined by the suppression and diversity introduction mechanisms. 3) The number of newcomers increases as the population increases in size. This is because if the population is continuously increasing in size, it is an indication that the problem has many local optima and the more optima it locates, the more it is capable of locating. 4) No encoding of the individuals of the population is required. 5) The associated computational cost of each iteration is $O(NL)$ or $O(N^2)$, where $N$ is the current population size and $L$ is the length of each vector. While performing only local search, Steps 2.1 to 2.5, the cost is $O(NL)$, and when interacting the network, Step 2.7, the cost is $O(N^2)$.

The affinity proportional mutation of Step 2.3 is performed according to the following expression:

$$c' = c + \alpha\, N(0,1), \qquad (1)$$
$$\alpha = (1/\beta)\exp(-f^*),$$

where $c'$ is a mutated cell $c$, $N(0,1)$ is a Gaussian random variable of zero mean and standard deviation $\sigma = 1$, $\beta$ is a parameter that controls the decay of the inverse exponential function, and $f^*$ is the fitness of an individual normalized in the interval [0,1]. A mutation is only accepted if the mutated cell $c'$ is within its range of domain.

Fig. 1 depicts the affinity proportional function $\alpha$ for the default value $\beta = 100$ used in the experiments reported here. Note that the mutation ratio from the best individual of the population to the worst one is approximately ½. Empirical results demonstrated that for large differences in the mutation rate (increase in the function decay) the worst individuals suffered a mutation much bigger than the best ones, resulting in the loss of the lower local optima solutions. In practical applications, some low local optima solutions might be the result of noise or imprecision on the modeling process. Thus, the possibility of adjusting the decay of the exponential function becomes an interesting approach to controlling the sensitivity of the search for global/local optima.

The stopping criterion adopted for the algorithm is based upon the size of the memory population. After network suppression, a fixed number of cells remains. If this number does not vary from one suppression to another, then the network is said to have stabilized and the remaining cells are memory cells corresponding to the solutions of the problem.

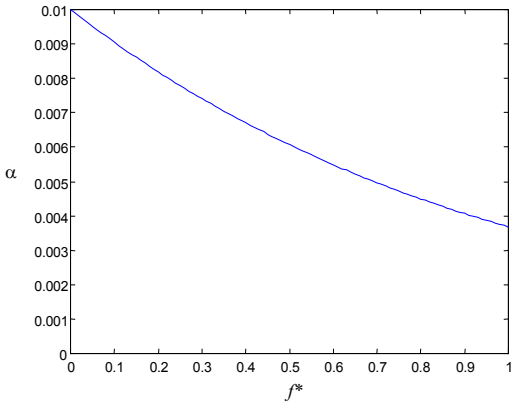



**Figure 1:** Function that performs the affinity proportionate mutation.

Although this strategy seems rather empirical, simulation results demonstrated its effectiveness for the problems tested. In any case, a pre-defined number of iteration steps can be adopted as an alternative.

## IV. RELATED STRATEGIES: SOME THEORETICAL ASPECTS

There is a number of differences between opt-aiNet and CLONALG. While CLONALG encodes the individuals of the population using binary strings similarly to a genetic algorithm, opt-aiNet is based upon real-valued vectors. The opt-aiNet includes the interaction of the network cells with the environment (fitness) and with each other (affinity), allowing the dynamic control of the population size. In the opt-aiNet case, newcomers are only allowed to enter the population after the current cells cannot significantly improve their average fitness. In CLONALG, the affinity proportionate mutation is based upon a control strategy suggested in [15] where short bursts of high mutation rates are followed by some breathing periods. In contrast, opt-aiNet follows a Gaussian mutation that is inversely proportional to the normalized fitness of each parent cell.

Opt-aiNet also presents a number of similarities with the evolution strategies (ES) introduced in [16]. Both use real-valued vectors to represent the individuals of the population. The selection mechanism of opt-aiNet can be equated to the $(\mu + \lambda)$-ES, where $\mu$ parents generate $\lambda$ offspring which are reduced again to $\mu$ parents. The selection operates on the joined set of parents and offspring, and parents can survive until they are superseded by one of their offspring. In opt-aiNet, $\mu = N$ (the whole population) and $\lambda = Nc$. Both strategies employ Gaussian mutation, with the difference that opt-aiNet has an affinity proportional Gaussian mutation with fixed standard deviation, while ES might use a fixed or time variant standard deviation and is not proportional to fitness. Another major difference between the two strategies is that opt-aiNet has a dynamic adjustment of the population size through metadynamics (diversity introduction) and network suppression, while ESs have a static number of individuals in the population.

## V. SIMULATION RESULTS

The opt-aiNet algorithm was applied to several uni- and bi-dimensional functions in order to assess its performance. The results reported in this paper illustrate its behavior for some of the problems tested and compare it with those results obtained by CLONALG. Three functions were tested:

1) Multi: function with several local optima solutions and a single global optimum all distributed non-uniformly.
2) Roots: function with six global optima and a large plateau.
3) Schaffer's: function with an infinite number of local optima and a single global optimum. The training parameters chosen for the opt-aiNet were the same in all cases:

- Suppression threshold: $\sigma_s = 0.2$;
- Initial number of network cells: $N = 20$;
- Number of clones generated for each cell: $Nc = 10$;
- Percentage of newcomers: $d = 40\%$;
- Scale of the affinity proportional selection: $\beta = 100$;
- Maximum number of iterations allowed: $Ngen = 500$.

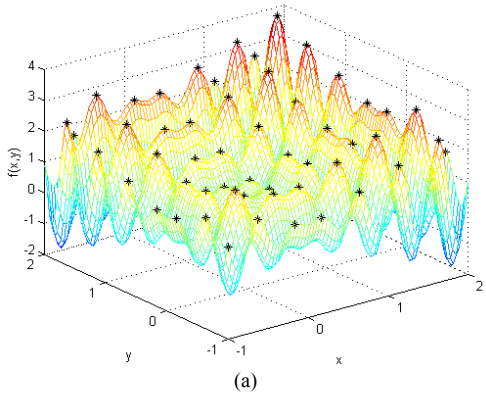
For the CLONALG, the following parameters were chosen: $Ngen = 200$, $n = N = 100$, $d = 10$, $\beta = 0.1$ (see [7] for a description of the parameters).

### A. Multi Function

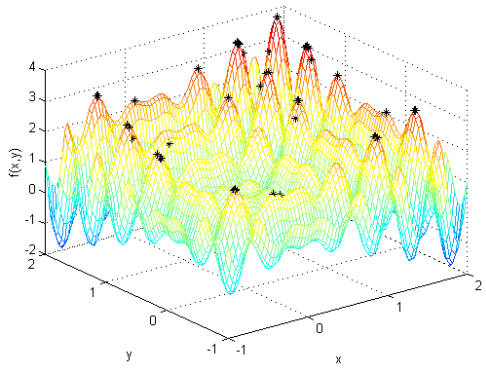
This function was used in [7] to evaluate the performance of CLONALG when applied to multimodal optimization and the results compared to that of a GA with fitness sharing. The authors claimed that CLONALG demonstrated to be capable of locating and maintaining a larger number of optima solutions than a GA with sharing. Fig. 2 illustrates the performance of CLONALG and opt-aiNet when applied to the Multi function (Eq. (2)). The opt-aiNet located 61 peaks, while CLONALG located only 18 peaks. Most importantly, the opt-aiNet positioned a single individual in each peak, avoiding the "waste of resources" presented by CLONALG.

$$g(x,y) = x.sin(4\pi x) - y.sin(4\pi y + \pi) + 1, \quad x,y \in [-2,2] \qquad (2)$$

Fig. 3 presents the average and the best fitness of the population for both algorithms. In this case the average fitness of CLONALG was larger than the average of opt-aiNet, indicating that it privileged the highest peaks of the function, in contrast to the opt-aiNet that tried to locate all the optima solutions. The opt-aiNet converged at iteration 451 according to the proposed stopping criterion.

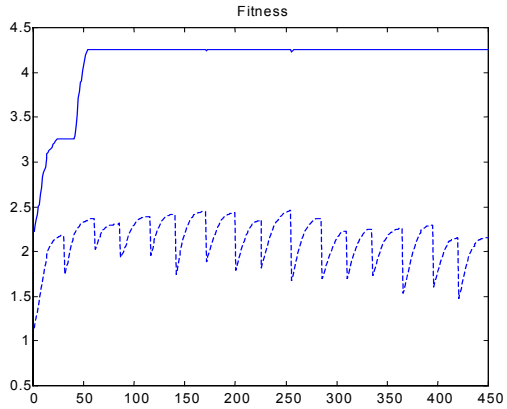



(a)

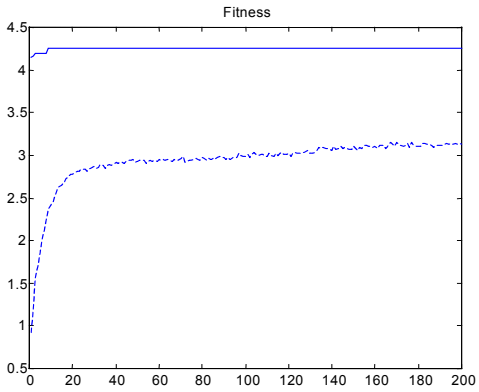



(b)

**Figure 2:** The Multi function. (a) opt-aiNet. (b) CLONALG.




(a)




(b)

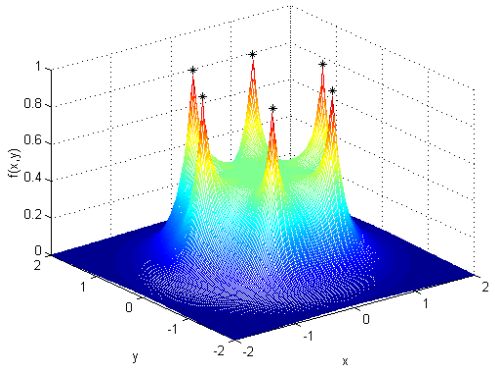
**Figure 3:** Fitness of the population. Best (solid line) and average (dashed line). (a) opt-aiNet. (b) CLONALG.

### *B. Roots Function*

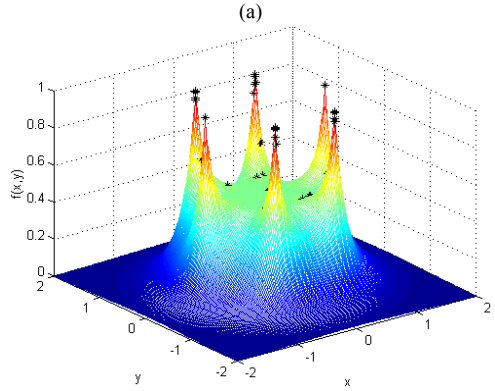
The Roots function is defined by Eq. (3). It takes its maxima at the six roots of units in the complex plane [17]. It presents a large plateau at the height 0.5, centered at coordinates (0,0) and surrounded by six thin peaks at height 1.0.

$$g(z) = \frac{1}{1+|z^6-1|}, \tag{3}$$

where $z \in \mathbf{C}$, $z = x + \mathrm{i}y$, $x,y \in [-2,2]$.

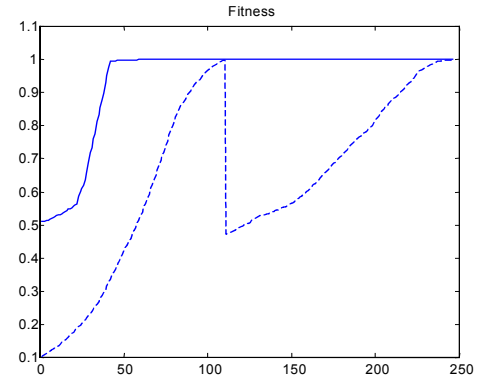



(a)

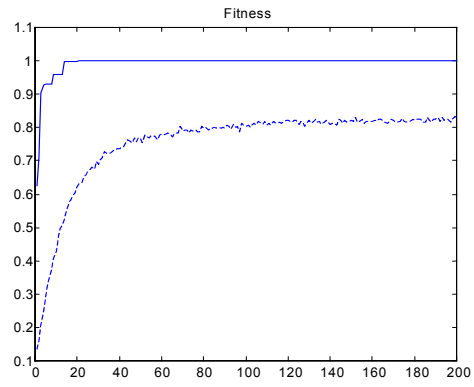



(b)

**Figure 4:** The Roots function. (a) opt-aiNet. (b) CLONALG.




(a)




(b)

**Figure 5:** Fitness of the population. Best (solid line) and average (dashed line). (a) opt-aiNet. (b) CLONALG.

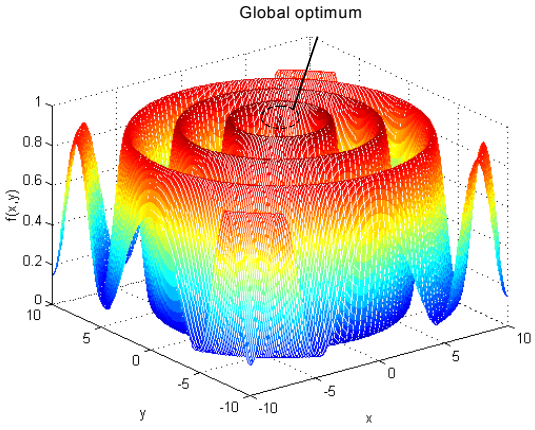
Fig. 4 presents the results of opt-aiNet and CLONALG when applied to the Roots function. In this case, the opt-aiNet converged after 246 iterations and was capable of locating all the peaks of the function with a final population of only six cells. The CLONALG also succeeded in locating all the optima of the Roots function, but it lacks a mechanism to define a more adequate number of individuals in the population. Fig. 5 presents the best and average error of the population for both algorithms. Note that the opt-aiNet optimizes locally the current population before inserting diversity, when the average fitness suffers a reduction, and starts the local optimization process again. After the second network suppression, the opt-aiNet maintained only six individuals as memory cells and converged.

### C. *Schaffer's Function*

The function used in [18] to study global optimization is described by Eq. (4). This function has a single global optimum at (0,0), $f(x,y) = 1$, and a large number of local optima. The global optimum is difficult to find because the value at the best local optimum differs from only about $10^{-3}$. As the local optima are not punctual, they form crowns around the global optimum, there are in fact an infinite number of local optima that form a sort of a trap around the global optimum. Fig. 6 illustrates the Schaffer's function in bi- and uni-dimensional plots. Note the subtle difference between the global optimum and the local optima.

$$g(z) = 0.5 + \frac{\sin^2\left(\sqrt{x^2+y^2}\right) - 0.5}{\left(1 + 0.001(x^2+y^2)\right)}, \quad x,y \in [-10,10] \qquad (4)$$

Fig. 7 presents the simulation results for opt-aiNet and CLONALG. Both algorithms were capable of locating the global maximum of the function. One difference is that the opt-aiNet performs a better exploration of the search space. The individuals of the population are more uniformly spread over the surface. In Fig. 8 it can be observed that CLONALG determined the global optimum faster than opt-aiNet and also, that the opt-aiNet did not converge for the 500 iterations. This means its population was still increasing in size and trying to locate a very large (infinite) number of local optima solutions.



(a)

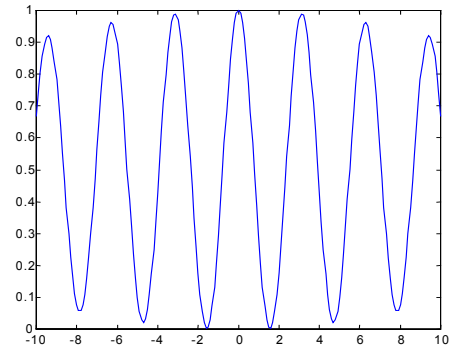
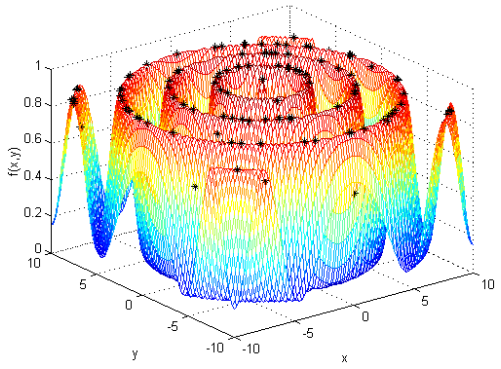
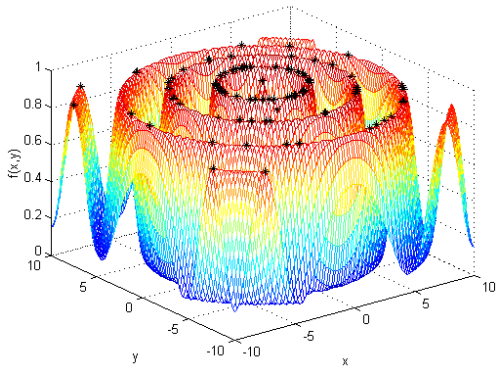

(b)

**Figure 6:** The Schaffer's function. (a) Bi-dimensional plot. (b) Uni-dimensional plot.

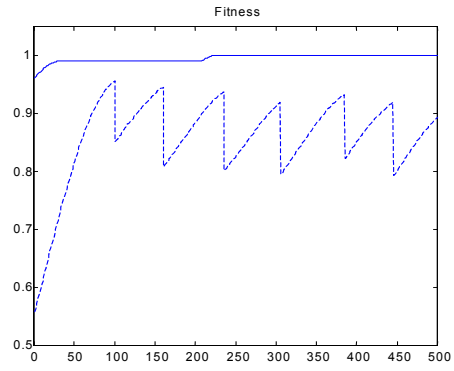


(a)

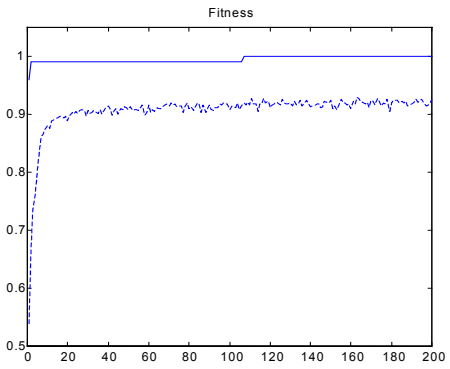


(b)

**Figure 7:** Simulation results for the Schaffer's function. (a) opt-aiNet. (b) CLONALG.



(a)




(b)

**Figure 8:** Fitness and final number of individuals in the population. Best (solid line) and average (dashed line) fitness. (a) opt-aiNet. (b) CLONALG.

## VI. DISCUSSION

Both algorithms are stochastic in nature. Each time they are run a different result is obtained. The performances illustrated in Figs. 2 to 8 can be said to be typical for the problems evaluated. However, it is important to evaluate the average performance of each algorithm. Table 1 summarizes the behavior of the algorithms when applied to the Multi and Roots functions. The values presented are the average and standard deviation taken over 10 runs of the algorithms. The number of peaks located by each algorithm is presented. In the opt-aiNet case, the number of iterations for convergence is also depicted. ItG corresponds to the number of iterations performed until the algorithms were capable of locating at least one global optimum of the function. Note that the opt-aiNet, on average, requires a higher number of iterations to locate the global optimum. This result is expected mainly because as the population grows dynamically in opt-aiNet, the algorithm started with a small initial population of size 20, five times smaller than the CLONALG population with 100 individuals. Nevertheless, the opt-aiNet is still capable of locating a larger number of optima solutions, automatically defining a stopping iteration and population size. No results are presented for the Schaffer's function because it has an infinite number of local optima.

As future trends for the optimization version of aiNet, several aspects still have to be accounted for. First, the algorithm sensitivity to its tuning parameters must be assessed. It has to be applied to real world and deceptive problems. It would also be interesting to compare the performance of both algorithms to any niching or crowding technique for a GA.

**Table 1:** Average performances for CLONALG and opt-aiNet when applied to the problems Multi and Roots. Peaks: number of peaks determined; ItG: number of iterations to locate the global optimum; ItC: number of iterations for convergence.

| | **CLONALG** | | **opt-aiNet** | | |
|---|---|---|---|---|---|
| | Peaks | ItG | Peaks | ItG | ItC |
| **Multi** | 31.6±3.53 | 43.1±19.84 | 56.10±4.36 | 53.50±57.19 | 278.50±70.09 |
| **Roots** | 6±0 | 23.30±5.50 | 6±0 | 86.89±34.31 | 295.00±129.74 |

## VII. CONCLUDING REMARKS

This paper presented a modified version of an artificial immune network model specially designed to solve multimodal optimization problems. It was theoretically compared with a clonal selection algorithm also applied to perform multimodal optimization, and evolution strategies.

The modified algorithm was presented and its performance illustrated for three bi-dimensional functions. The algorithm demonstrated to be capable of combining exploitation with exploration of the fitness landscape and showed a good stabilization of the population. Its search method is based on local search intertwined with global search. Other important features of the algorithm are dynamic search for an optimum population size based upon the network suppression threshold and a well-defined stopping criterion.

### Acknowledgements

Leandro N. de Castro thanks the Computing Laboratory at UKC for the financial support.

### References

[1] Ishida, Y., Hirayama, H., Fujita, H., Ishiguro, A. and Mori, K. (eds.) (1998), *Immunity-Based Systems--Intelligent Systems by Artificial Immune Systems*, Corona Pub. Co. Japan (in Japanese).

[2] Dasgupta, D. (ed.) (1999), *Artificial Immune Systems and Their Applications*, Springer-Verlag.

[3] de Castro, L. N. & Timmis, J. (2002), An Introduction to Artificial Immune Systems: A New Computational Intelligence Paradigm, Springer-Verlag.

[4] Perelson, A. S. & Oster, G. F. (1979), "Theoretical Studies of Clonal Selection: Minimal Antibody Repertoire Size and Reliability of Self-Nonself Discrimination", *J. theor.Biol.*, 81, pp. 645-670.

[5] Forrest, S., A. Perelson, Allen, L. & Cherukuri, R. (1994), "Self-Nonself Discrimination in a Computer", *Proc. of the IEEE Symposium on Research in Security and Privacy*, pp. 202-212.

[6] Seiden, P. E. & Celada, F. (1992), "A Model for Simulating Cognate Recognition and Response in the Immune System", *J. theor. Biol.*, 158, pp. 329-357.

[7] de Castro, L. N., & Von Zuben, F. J., (2001), "Learning and Optimization Using the Clonal Selection Principle", IEEE Trans. on Evol. Comp., Special Issue on Artificial Immune Systems (in print).

[8] Farmer, J. D., Packard, N. H. & Perelson, A. S. (1986), "The Immune System, Adaptation, and Machine Learning", *Physica 22D*, 187-204.

[9] Varela, F. J. & Coutinho, A. (1991), "Second Generation Immune Networks", *Imm. Today*, 12(5), pp. 159-166.

[10] de Castro, L. N. & Von Zuben, F. J. (2001), "aiNet: An Artificial Immune Network for Data Analysis", in *Data Mining: A Heuristic Approach*, H. A. Abbass, R. A. Sarker, and C. S. Newton (eds.), Idea Group Publishing, USA, Chapter XII, pp. 231-259.

[11] Timmis, J. (2000), *Artificial Immune Systems: A Novel Data Analysis Technique Inspired by the Immune Network Theory*, Ph.D. Dissertation, Department of Computer Science, University of Wales.

[12] Ada, G. L. & Nossal, G. J. V. (1987), "The Clonal Selection Theory", *Scientific American*, 257(2), pp. 50-57.

[13] Jerne, N. K. (1974), "Towards a Network Theory of the Immune System", *Ann. Immunol.* (Inst. Pasteur) 125C, pp. 373-389.

[14] de Castro, L. N. & Von Zuben, F. J., (2001), "Immune and Neural Network Models: Theoretical and Empirical Comparisons", *Int. Journal of Comp. Intelligence and Applications*, 1(3), pp. 239-257.

[15] Kepler, T. B. & Perelson, A. S. (1993), "Somatic Hypermutation in B Cells: An Optimal Control Treatment", *J. theor. Biol.*, 164, pp. 37-64.

[16] Schwefel, H. –P. (1965), *Kybernetische Evolutionals Strategie der Experimentellen Forschung in der Stromungstechnik*, Diploma Thesis, Technical University of Berlin.

[17] Pétrowski, A. & Genet, M. G. (1999), "A Classification Tree for Speciation", *Proc. of the CEC'99*, pp. 204-211.

[18] Schaffer J. D., Caruana, R. A., Eshelman, L. J. & Das, R. (1989), "A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization", *Proc. of the 3rd Int. Conference on Genetic Algorithms*, pp. 51-60.