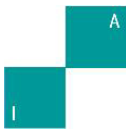
Inteligencia Artificial 46(2010), 3-16
doi: 10.4114/ia.v14i46.1500



INTELIGENCIA ARTIFICIAL

http://erevista.aepia.org/

Artificial Immune System for Solving Global Optimization Problems

Victoria S. Aragón[1], Susana C. Esquivel[1], Carlos A. Coello Coello[2]

[1] Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
Universidad Nacional de San Luis
Ejército de los Andes 950
(5700) San Luis - Argentina
{vsaragon, esquivel}@unsl.edu.ar

[2]CINVESTAV-IPN (Evolutionary Computation Group)
Electrical Eng. Department, Computer Science Dept.
Av. IPN No. 2508, Col. San Pedro Zacatenco
México D.F. 07300, MÉXICO
ccoello@cs.cinvestav.mx

Abstract In this paper, we present a novel model of an artificial immune system (AIS), based on the process that suffers the T-Cell. The proposed model is used for global optimization problems. The model operates on four populations: Virgins, Effectors (CD4 and CD8) and Memory. Each of them has a different role, representation and procedures. We validate our proposed approach with a set of test functions taken from the specialized literature, we also compare our results with the results obtained by different bio-inspired approaches and we statistically analyze the results gotten by our approach.

Keywords: Artificial Immune System, Global Optimization Problems.

1 Introduction

Problem optimization has been an active area of research. As the real world problems are more complex the algorithms for solving them have to be faster and accurate.

Over the last years, a bio-inspired system has call the attention of some researchers, the Natural Immune System (NIS) and its powerful capacity of information processing [27]. The NIS is a very complex system with several defense mechanisms against foreign organisms. The main purpose of the NIS is recognize all cells of the host and categorize them in order to induce the appropriate immune response. The NIS learns through the evolution to distinguish between self and non-self. The NIS has many desirable characteristics from the point of view computational, such as: uniqueness, pattern recognition, diversity, tolerance faults, learning and memory, self-organization, robustness, cooperation between different layers, among others. Thus, these characteristics and a well-known functionality about the NIS are excellent motivations to develop Artificial Immune Systems (AIS) to hand global optimization problems.

Two theories which motivated the most popular immune-inspired models in current use are : Clonal Selection Theory and Immune Network Theory. The Clonal Selection principle is the theory used to describe the basic properties of an adaptive immune response to an antigenic stimulus [27]. The Clonal Selection's premise is that an adaptive immune system reacts only when it is invaded by an external stimulus (i.e., an antigen). Thus, the adaptive immune system reaction or response is determined through the production of antibodies by the B-cells, once the

ISSN: 1988-3064(on-line)

antigen arrives into the system. So, in the presence of an antigen, those antibodies with the highest recognizing level are selected to proliferate, producing huge volumes of antibodies, called clones. The proliferation is done through an asexual procedure. During this reproduction stage, the clones are subject to somatic hypermutation and, with a strong selection pressure, B-cells are obtained with high affinity receptors for this particular antigen. By the other side, the Immune Network Theory's premise is that the antibodies (B-cell receptors) have some portions called *idiotopes*, which can be recognized by other free antibodies or receptor molecules on the B-cells. This capability of recognizing and being recognized by other elements of the immune system gives to the system an intrinsic dynamic behavior [27]. N. K. Jerne proposed in [20] the philosophical issues related to the behavior of immune networks, but he did not present a formal model. One of the open issues back then was how to deal with an interaction between two elements, that is, suppressive or stimulatory. In 1974, Jerne [21] proposed a pioneering model of the immune network. Currently, there are both continuous and discrete immune network models. For the purposes of this paper, only the discrete model is considered.

The discrete models are based on differential equations and/or interactive procedures of adaptation that govern the network's behavior. These types of models were designed to solve problems and they interact with an external environment. Besides, they adapt the number and structure of their elements, which means that the cells and molecules can increase or decrease their number and shape (these are attributes in the shape-space) in order to improve their affinity.

According to the Immune Network Theory proposed by Jerne [20], the immune system is composed by a network whose dynamics can be disturbed by foreign antigens. These interactions with the antigens yield a network that corresponds to an internal image of the universe of antigens. Into the network, groups of cells and molecules correspond to groups of antigens. There are two levels of interaction, with the environment and with other network elements [27].

The main motivation of the work presented in this paper is to verify the behavior of a new model of artificial immune system, it is called CTM, in the context of global optimization, the algorithm solves some function minimization problems taken from the specialized literature. This model is based on the process that suffers the T-Cell. We decide to propose a new model in order to exploit different characteristics of the NIS still not much studied, such as, the secretion of cytokines by the T cells for communication purposes and to reflect the phase T cells undergoing from virgin cells to they become memory cells.

Taking into account the premises each model is based on, the Clonal Selection Theory says that the immune system reacts when it is invaded by an antigen proliferating and hypermutating the best cells with a strong selection pressure. The Immune Network Theory afirms that lymphocytes interact between them with or without an external invasion, also proliferating and hypermutating the best cells. In contrast, under CTM, the T cells react in the presence of an antigen plus co-stimulating signals sent by the T cells, according to the Danger Theory [12].

CTM involves, into its recognizing phase, the self/nonself concept (negative selection [13]) in order to eliminate those cells that do not cooperate in the immunitary response or that could be dangerous for the host. Thus, while the negative selection model is oriented towards the generation of a set of detectors, CTM adopts the concept of negative selection in order to remove cells that are too similar with the purpose of mantaining diversity.

CTM considers the interaction process among T cells through the secretion of cytokines, according to the cytokine network model [12]. Interactions in the discrete Immune Network models are given by differential equations and/or interactive procedures of adaptation. Clonal Selection Theory does not include interactions among its cells.

The remainder of the paper is organized as follows. In Section 2, we define the problem we want to solve. Section 3 describes some previous related work. In Section 4, we introduce the approach. In Section 5, we present our experiments. In Section 6, our results are presented and they are discussed. Finally, in Section 7, we present our conclusions and some possible paths for future work.

2 Statement of the Problem

A global optimization problem can be formalized as a pair (S, f) where $S \subseteq R^n$ and $f :\rightarrow R$ is an n-dimensional real-value function. The goal is to find (if we are minimizing) a $\vec{x}^* = (x_1, \ldots, x_n) \in S$ such as $\vec{x}^*$ is a global optimum (minimum) on S, i.e. $\forall \vec{x} \in S : f(\vec{x}^*) \leq f(\vec{x})$. This generic function can be difficult to optimize due to the presence of many local optima. In general, the difficulty increases when the dimensionality of the problem is growing up.

3 Previous Work

Fast Evolutionary Programming (FEP) is one of the best evolutionary algorithms for numerical optimization. FEP is based on Conventional Evolutionary Programming and it uses a mutation operator based on Cauchy random numbers to escape from local optima. FEP was validated with 23 functions taken from the specialized literature [36].

Conventional Evolutionary Programming (CEP) is a conventional evolutionary programming which uses three different mutation operators: a Gaussian Mutation Operator (GMO), a Cauchy Mutation Operator (CMO) and a Mean Mutation Operator (MMO). CEP was validated with 11 functions taken from the specialized literature [2].

The clonal selection theory has inspired many algorithms, Brownlee in [1] defined a lineage for this kind of algorithms. The Artificial Immune Recognition System (AIRS) is a supervised learning algorithm for classification problems. It uses the idea of Artificial Recognition Ball (ARB) [28] to represent clones of identical B cells. Differents works about AIRS can be found in [33, 34, 31, 32].

The B-Cell Algorithm (BCA)[22, 23, 29, 19] was applied to optimization problems, the Clonal Selection Algorithm (CLONALG)[15, 16] is the most popular algorithm based on the clonal selection theory. Different clonal selection algorithms are described next.

Yidan Luo and Zhongyang Jiang proposed in [25] an improved clonal selection algorithm (ICSA) to deal with the problem of easily being trapped in local minima and slow convergence of clonal selection algorithm. The improved algorithm included orthogonal crossover, simplex crossover, clone and selection. The idea of evolutionary computation was integrated into clone selection and the authors proposed a new mutation operator. The authors argue the new algorithm can guarantee the diversity of the population and improve the global search ability. Three functions were utilized to test this method and the simulation results have shown that the proposed ICSA algorithm has good performance.

Maoguo Gong et. al proposed in [18] to introduce chaos into the Clonal Selection Algorithm by a novel mutation method, self-adaptive chaotic mutation (SACM). Based on the logistic chaotic sequence, SACM extracts antibody's affinity and distribution to adjust the mutation scale. They compared SACM with the Clonal Selection Algorithm using random mutation and Standard Genetic Algorithm over 11 global functions, the authors argue the SACM can enhance the precision and stability, and overcome prematurity effectively with a high convergence speed.

The Immunological Algorithm family (IA) [8, 11, 6, 7, 3, 5, 10, 9] is another kind of clonal selection algorithms.

Olivetti de França F., Von Zuben F. and Nunes de Castro L. propose a multimodal optimization algorithm inspired by the immune human system, it is called OPT-AINET. It encoded the solutions with real values in an Euclidean shape-space, OPT-AINET is based on the immune network. OPT-AINET was validated with 18 functions taken from the specialized literature [17].

Cutello V., Narzisi G., Nicosia G. and Pavone M. propose an immunological algorithm for continuous global optimization problems called OPT-IA. It is based on the clonal selection principle. OPT-IA uses a cloning operator, an inversely proportional hypermutation operator and an aging operator, in order to eliminate the oldest cells or solutions. OPT-IA was validated with 23 functions taken from the specialized literature [4].

Cutello V. and Nicosia G. and Pavone M. propose an improved version of OPT-IA [4] called OPT-IMMALG. This approach uses real-code representation and an inversely proportional hypermutation operator. OPT-IMMALG was validated with 23 functions taken from the specialized literature [11] .

Jie Yang et. al proposed in [35] a novel artificial immune algorithm, Lamarckian Clonal Selection Algorithm(LCSA), to solve some benchmark problems. This algorithm combines different properties derived from CSA and local search technique, the authors argue it can perform well on exploration and exploitation. From the analysis and experiments, the authors can find the validity while using Lamarckian learning in the clonal selection algorithm, which makes the search more definite. The authors make some suggestions to improve search efficiency and reduce complexity.

Li Liu et. al proposed in [24] a cooperative artificial immune network denoted CoAIN for multimodal function optimization. To explore and exploit searching space efficiently and effectively, the interactions within the network are not only suppression but also cooperation. Network cells cooperate with particle swarm behavior making use of the best position encountered by itself and its neighbor. Numeric benchmark functions were used to assess the performance of CoAIN compared with opt-aiNet, BCA, hybrid GA, and PSO algorithms. CoAIN presented a good performance, in terms of optimal objective function values, number of function evaluations, and average run time. The authors argue that it indicates that some basic immune principles together with simple cooperation behavior

can help them to solve complex optimization tasks. The authors raise artificial immune network of CoAIN needs to be improved its performance for optimization in higher dimensions and dynamic environments.

Dasgupta S. et. al proposed in [14] a micro-bacterial foraging optimization algorithm, which evolves with a very small population compared to its classical version. In this modified bacterial foraging algorithm, the best bacterium is kept unaltered, whereas the other population members are reinitialized. This new small population μ-BFOA is tested over five numerical benchmark problems for high dimensions (500) and the authors find this to outperform the normal bacterial foraging with a larger population as well as with a smaller population.

4 OUR PROPOSED MODEL

This paper presents a novel bio-inspired model, it is called CTM. It is a new model of adaptive immune system based on the immune responses mediate by the T-cell. Its premise is that the T-cells only react with the presence of an antigen plus co-stimulant signals, through a series of actions. These actions are influenced by a set of signal emitted by the T-cells, i.e., the signal determine the level which the actions are trigger: proliferation and differentiation of the T-cells.

This model operates on four populations, corresponding to the four groups in which the T-cells are divided: (1) Virgin Cells (VC), (2) Effector Cells with cluster denomination CD4 (CD4), (3) Effector Cells with cluster denomination CD8 (CD8) and (4) Memory Cells (MC). The cluster denomination determines the properties of the cells. Each population is composed by a set of T-cells whose characteristics are subject to the population which belong to.

CTM consists on two phases, the first (*recognizing phase*) is about the processes that suffer only the virgin cells and the second one (*effector phase*) is related to the processes that suffer the cells in CD4, CD8 and MC. The *recognizing phase* has to provide two populations (CD4 and CD8) with some diversity in order to the next phase can find a cell to optimize the given problem. While, the *effector phase* is in change to find this cell.

The characteristics of each cell and the processes that suffers are the following.

- Virgin Cells (VC): these cells do not suffer the activation process, i.e., they do not proliferate (clonal selection) nor differentiate. The VC's function inside the model is to provide diversity. This is reached through the random acquisition of TCR receptors. The T-Cell Receptor (TCR) can be represented by bit strings or strings of real value. In the natural immune system, the positive selection and negative selection have as goal to eliminate those cells that do not cooperate or could be dangerous for the host. Taking into account this concept, the cells in the model are exposed to these selections. The positive selection discards those cells with a low recognizing level to the antigen. The negative selection discard the similar cells (according to a threshold) in order to mantain diversity in the population. The virgin cells are represented by:
  - A *TCR* (TCR_b): represented by bit strings using Gray coding, it identifies the decision variables of the problem.
  - A *TCR* (TCR_r): represented by a string of real values, it identifies the decision variables of the problem.
  - A cluster denomination *CD4*: if it is active then the valid TCR is TCR_b.
  - A cluster denomination *CD8*: if it is active then the valid TCR is TCR_r.
  - Objective function values for TCR_b.
  - Objective function values for TCR_r.

  At the beginning both cluster denominations are active. Before the positive selection only one of them will be active, the one that shows the best recognizing to the antigen.
- Effector Cells with cluster denomination CD4: these cells suffer the activation process. The goal of these kind of cells is to explore the conflicting regions of the search space using the properties of the bit strings Gray coding representation. A cell from CD4 is composed by:
  - A *TCR* (TCR_b): represented by a bit string using Gray coding, it identifies the decision variables of the problem.
  - Objective function values for TCR_b.

  - Proliferation Level: it indicates the number of clones that will be receive the cell.
  - Differentiation: it indicates the number of bits that will be change, when the differentiation process is applied.

- Effector Cells with cluster denomination CD8: these cells suffer an activation process. The goal of these kind of cells is the same that CD4. But they use the properties of the real values encoded. A cell from CD8 is composed by:
  - A *TCR* (TCR_r): represented by a string of real values, it identifies the decision variables of the problem.
  - Objective function values for TCR_r.
  - Proliferation Level: it indicates the number of clones that will be receive the cell.
  - Differentiation: it indicates the number of decision variables that will be change, when the differentiation process is applied.

- Memory Cells (MC): these cells too suffer the activation process. The goal of these kind of cells is to explore the neighborhood of the best found solutions employing the real value representation. These cells are represented by the same components that CD8, but they suffer different processes.

- The activation of an effector cell, called ce_i, implies the random selection of a set of potential activator (or stimulating) cells. The closer one to ce_i, using Hamming or Euclidean distance according to the TCR, in the set is chosen to become the stimulating cell, say ce_j. Then, ce_i proliferates and differentiates.

- At the beginning, the proliferation level of each stimulated cell, ce_i, is given by a random value between [1, $\mid VC \mid$], but then it is determined taking into account the proliferation level of its stimulating cell (ce_j). If the ce_i is better than ce_j, then ce_i keeps its own proliferation level; otherwise, ce_i receive a level which is 10% lower than level of ce_j.

- Memory cells proliferate and differentiate according to their proliferation and differentiation levels, respectively. Both levels of a memory cell are independent from the others memory cells.

- Exist a comunication process between CD4 and CD8, where the best cell from CD4 replaces the worst cell from CD8. As the representation of the TCR, for CD4 and CD8, are different, before the insertion of the TCR_b into CD8 it has to be converted in its real-value vector (TCR_r), for this process we use the following equation, it takes a bitstring using Gray coding and returns a real number (this process is applicable as many times as decision variables has the problem):

$$dv_j = l_{lj} + \frac{\sum_{i=0}^{L_j} 2^{L_j - i} dv'_{ij}(lu_j - ll_j)}{2_j^L - 1} \tag{1}$$

  where dv_j is the j^{th} decision variable with $j = 1, \ldots,$ number of decision variables of the problem, L_j is the number of bits for the j^{th} decision variable, lu_j and ll_j are the upper and lower limits for the decision variable dv_j and dv'_{ij} is the i^{th}-bit of the binary string that represents dv_j.

- Each type of cell has its own differentiation process, which is blind to their representation and population.

  Differentiation for CD4 : the differentiation level of ce_i is determined by the Hamming distance between the stimulated (ce_i) and stimulating (ce_j) cells. It indicates the number of bits to be changed. Each decision variable and the bit to be invested are chosen in a random way. The bit changes according to a mutation (or reaction) probability $prob_{mut-CD4}$.

  Differentiation for CD8 : the differentiation level for cell ce_i is a random between [1, $\mid dv \mid$/ 2], where $\mid dv \mid$ is the number of decision variables of the problem. Each variable to be change is chosen in a random way and it is modified according to the following equation:

$$x^{'} = x \pm \left(\frac{U(0, lu - ll)}{10000000 iter} \right)^{U(0,1)} \tag{2}$$

Algorithm 1 Pseudo-code for CTM

```
Initialize_VC();
Evaluate_VC();
Assign_Proliferation();
Active_CDs();
Divide_CDs();
Positive_Selection_CD4();
Positive_Selection_CD8();
Negative_Selection_CD4();
Negative_Selection_CD8();
while Repeat a number of evaluations do
   while Repeat a number of times (rep_CD4) do
      Active_CD4();
   end while
   Sort_CD4();
   Comunication_CD4_CD8();
   while Repeat a number of times (rep_CD8) do
      Active_CD8();
   end while
   Sort_CD8();
   Insert_CDs_en_MC();
   while Repeat a number of times (rep_MC) do
      Active_MC();
   end while
   Sort_CM();
   Statistics();
end while
```

where x and $x^{'}$ are are the original and mutated decision variables, respectively. $U(0,w)$ refers to a random number with a uniform distribution between (0,w). lu and ll are the upper and lower bounds of x and $iter$ is the current iteration number. At the moment of the differentiation of a cell, it taking into account the value of objective function of its stimulating cell. In order to determinate if $r = \left(\frac{U(0,lu-ll)}{10000000iter}\right)^{U(0,1)}$, will be add or substrates to x, the following criterion are considered: 1) if the stimulating cell is better than the stimulated cell and the decision variable value of the first cell is less than the second one or if the stimulated cell is better than the stimulating cell and the decision variable value of the first cell is less than the second one then r is rested to x and 2) if the stimulating cell is worst than the stimulated cell and the decision variable value of the first cell is less than the second one or if the stimulating cell is better than the stimulated cell and the decision variable value of the first cell is larger than the second one then r is added to x. Both criterion are aimed to guide the search to the best found solutions.

Differentiation for MC : the number of decision variables that will be changed is given by the differentiation level, it is a random between [1, 10% $\mid dv \mid$], where $\mid dv \mid$ is the number of decision variables of the problem each variable is chosen in a random way and it changes according to:

$$x^{'} = x \pm \left(\frac{U(0,lu-ll)}{10000000iter}\right)^{U(0,1)} \tag{3}$$

where x and $x^{'}$ are are the original and mutated decision variables, respectively. $U(0,w)$ refers to a random number with a uniform distribution between (0,w). lu and ll are the upper and lower bounds of x and $iter$ is the current iteration number. In a random way we decide if $r = \left(\frac{U(0,lu-ll)}{10000000iter}\right)^{U(0,1)}$, will be added or subtracted to x.

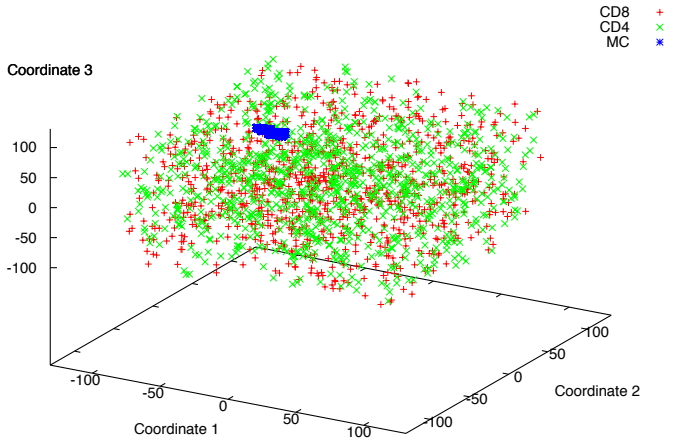


Figure 1: Comparison between distribution of 1000 mutated points generated by the three differentiation operators (CD4 - prob$_{diff-CD4}$=0.5 , CD8 and MC), on the same original point.

Therefore, the general structure of our proposed model for global optimization problems is given in Algorithm 1.

In order to corroborate if the differentiation operators works in the desirable way, we generate an original point three dimensional, in the range $[0, 100]$ and we apply, to all dimensions, the three operators to it 1000 times. For binary Gray conding (CD4) we use 41 bits to represent each decision variable and the number of bits to change was setting to 18 bits (differentiation level). Figure 1 shows the superposition of the generated points for each operator. We can see the distribution of CD8 points are similar to CD4 points with prob $_{diff-CD4}$=0.5, it means a good global search. Besides, it is very clear how the operator for MC performs a deep local search.

5 Experimental Setup

In order to validate our proposed model we tested it with a benchmark of 23 test functions taken from the specialized literature [36] (see Table 1). The functions can be divided on three groups with different degree of difficulty: unimodal functions (f_1 to f_7) which are relatively easy to optimize but their complexity increases with the dimensionality; multimodal functions (f_8 to f_{13}) they have many local optima and they are hard to be solving by some optimization algorithms; multimodal functions (f_{14} to f_{23}) with a few local optima. Note that f_6 is a discontinuous step function with only one optimum and f_7 is a noise function which involves an uniformly distributed random variable within $[0, 1]$. All problems are minimization problems. 50 independents runs were performed for each problem and the parameter settings are: $\mid VC \mid$=100 cells, $\mid CD4 \mid$=$\mid CD8 \mid$= $\mid MC \mid$ = 20 cells for all functions, except for f_2, f_3, f_4, f_8, f_9, f_{10} and f_{12} for them $\mid VC \mid$=10 cells, $\mid CD4 \mid$=$\mid CD8 \mid$= $\mid MC \mid$ = 5 cells. prob$_{mut-CD4}$ was fixed in 0.01 for f_1 to f_6, f_8, f_{10}, f_{12} and f_{23}, the remaining functions use prob $_{mut-CD4}$= 0.02. rep$_{CD4}$ was fixed in 800 for all functions, except for f_9 and f_{23}, they used 80 and 100 repetitions. rep $_{CD8}$=rep$_{MC}$= 10 for all functions. Table 2 shows the number of objective function evaluations for each problem.

Our results are indirectly compared with respect to different kind of bio-inspired approaches: a Differential Evolution algorithm (DE) [30], a Particle Swarm Optimizer (PSO) [30], a simple Evolutionary Algorithm (SEA) [30], an immunological algorithm for continuous global optimization problems (OPT-IA) [4] and an improved version of OPT-IA (OPT-IMMALG) [11]. OPT-IA and OPT-IMMALG use the same number of function evaluations that CTM. DE, PSO and SEA use 5×10^5 function evaluations for all test functions.

Table 1: The 23 test functions used in our experimental studies, where n is the dimension of the function, f_{min} is the minimum value of the function, and $S \in R^n$

Test Function	n	S	f_{min}
$f_1(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]^n$	0
$f_2(\mathbf{x}) = \sum_{i=1}^{n} \mid x_i^2 \mid + \prod_{i=1}^{n} \mid x_i^2 \mid$	30	$[-10, 10]^n$	0
$f_3(\mathbf{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	30	$[-100, 100]^n$	0
$f_4(\mathbf{x}) = max_i \left\{ \mid x_i^2 \mid, 1 \leq i \leq n \right\}$	30	$[-100, 100]^n$	0
$f_5(\mathbf{x}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]^n$	0
$f_6(\mathbf{x}) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	30	$[-30, 30]^n$	0
$f_7(\mathbf{x}) = \sum_{i=1}^{n} i x_i^4 + random\,[0, 1)$	30	$[-1.28, 1.28]^n$	0
$f_8(\mathbf{x}) = \sum_{i=1}^{n} -x_i sin\left(\sqrt{\mid x_i \mid}\right)$	30	$[-500, 500]^n$	-12569.5
$f_9(\mathbf{x}) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	30	$[-5.12, 5.12]^n$	0
$f_{10}(\mathbf{x}) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right)$ $+20 + e$	30	$[-32, 32]^n$	0
$f_{11}(\mathbf{x}) = \frac{1}{4000} + \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right)$	30	$[-600, 600]^n$	0
$f_{12}(\mathbf{x}) = \frac{\pi}{n}\left(10sin^2(\pi y_i) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10sin^2(\pi y_{i+1})\right](y_n - 1)^2\right)$ $+ \sum_{i=1}^{n} u(x_i, 10, 100, 4)$, $y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	30	$[-50, 50]^n$	0
$f_{13}(\mathbf{x}) = 0.1(sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + sin^2(3\pi x_{i+1})] + (x_n - 1)$ $[1 + sin^2(2\pi x_n)]) + \sum_{i=1}^{n} u\,(x_i, 5, 100, 4)$	30	$[-50, 50]^n$	0
$f_{14}(\vec{x}) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6 +}\right]^{-1}$	2	$[-65.536, 65.536]^n$	1
$f_{15}(\vec{x}) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	$[-5, 5]^n$	0.0003075
$f_{16}(\vec{x}) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1 x_2 + (-4 + 4x_2^2)x_2^2$	2	$[-5, 5]^n$	-1.0316285
$f_{17}(\vec{x}) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10$	2	$[-5, 10] \times [0, 15]$	0.398
$f_{18}(\vec{x}) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2$ $+6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2$ $-36x_1x_2 + 27x_2^2)]$	2	$[-2, 2]^n$	3.0
$f_{19}(\mathbf{x}) = -\sum_{i=1}^{4} c_i \cdot exp\left(-\frac{1}{\pi}\sum_{j=1}^{4}(x_j - a_{ij})^2\right) \cdot$	4	$[0, 1]^n$	-3.86
$f_{20}(\mathbf{x}) = -\sum_{i=1}^{4} c_i \cdot exp\left(-\frac{1}{\pi}\sum_{j=1}^{6}(x_j - a_{ij})^2\right) \cdot$	6	$[0, 1]^n$	-3.32
$f_{21}(\mathbf{x}) = -\sum_{i=1}^{5}[(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]^n$	-10.15
$f_{22}(\mathbf{x}) = -\sum_{i=1}^{7}[(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]^n$	-10.39
$f_{23}(\mathbf{x}) = -\sum_{i=1}^{1} 0[(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]^n$	-10.53

Table 2: Evaluations for each Problem

Function	Evaluations
f_1, f_5, f_6, f_{10}, f_{12}, f_{13}	150000
f_2, f_{11}	200000
f_3, f_4, f_9	500000
f_7	300000
f_8	900000
f_{14}, f_{v16}, f_{17}, f_{18}, f_{19}, f_{21}, f_{22}, f_{23}	10000
f_{15}	400000
f_{20}	20000

6 Discussion of Results

Table 3 shows the results obtained with CTM. We can see that CTM was able to reach the optimum in 13 of the 23 test functions (f_3, f_6, f_7, f_{12}, f_{14}, f_{16}, f_{17}, f_{18}, f_{19}, f_{20}, f_{21}, f_{22} and f_{23}). Besides, CTM was able to reach the optimum on the 50 runs in 11 of the 23 test functions (f_3, f_7, f_{12}, f_{14}, f_{16}, f_{17}, f_{18}, f_{19}, f_{20}, f_{21} and f_{22}). Additionally, it found acceptable (i.e., not too far from the global optimum) solutions for the rest of the test functions.

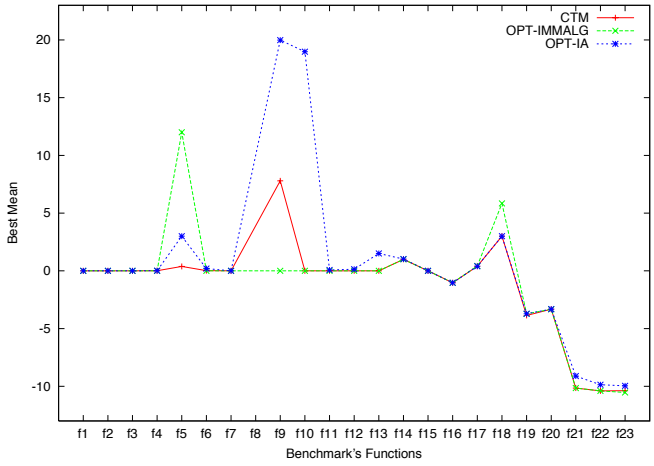
If we consider the groups in which the test functions are divided, we can see that CTM does not have an excellent performance under the easier functions (f_1 to f_7), only in three of the seven test function of this group CTM reaches the optimum. Under the second group (f_8 to f_{13}) CTM shows a good performance and for the last group (f_{14} to f_{23}), our proposed model has an excellent performance, only in one of the eleven functions CTM does not get the optimum.

Comparing CTM with respect to the two AIS (OPT-IMMALG and OPT-IA) (see Table 4), CTM and OPT-IMMALG get a similar performace, if we consider the number of test functions in which the approaches reach the optimum in all runs. OPT-IMMALG is better than CTM under the first and second groups of test functions and CTM is better under the third group. CTM, in general, overcome the performance of OPT-IA. Comparing CTM with respect to DE, PSO, SEA (see Table 4), DE is the only approach that shows a performance better than CTM but only for the first and second function groups. Figures 2a) and 2b) show the best mean obtained by 1) CTM, OPT-IMMALG and OPT-IA and 2) CTM, DE, PSO and SEA, respectively, for all test functions except f_8. Figure 2c) shows the mean obtained by all approaches for f_8.

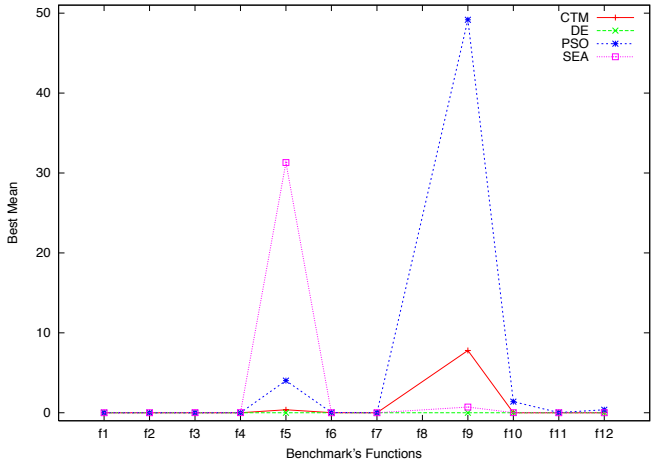
Table 3: Results obtained by CTM

Function	Optimum	*Best*	*Worst*	*Mean*	*St.Dev.*
1	0.0	1.0×10^{-10}	4.1×10^{-9}	7.0×10^{-10}	9.0×10^{-10}
2	0.0	4.45×10^{-8}	1.04486×10^{-5}	1.4591×10^{-6}	1.9639×10^{-6}
3	0.0	**0.0**	**0.0**	**0.0**	0.0
4	0.0	9.70431×10^{-5}	$5.21267229 \times 10^{-2}$	8.6612784×10^{-3}	9.0135040×10^{-3}
5	0.0	2.1935881×10^{-3}	2.7374691261	$3.892737636 \times 10^{-1}$	$6.100025859 \times 10^{-1}$
6	0.0	**0.0**	1.0	0.02	0.141421
7	0.0	**0.0**	**0.0**	**0.0**	0.0
8	-12569.5	-12569.4866181730	-12450.6911288609	-12540.9514628515	51.05865
9	0.0	2.05132×10^{-4}	21.8890734828	7.8122983770	5.4742058641
10	0.0	1.469×10^{-7}	5.52060×10^{-5}	6.4684×10^{-6}	8.5423×10^{-6}
11	0.0	1.340×10^{-7}	$1.32443213 \times 10^{-2}$	2.0773141×10^{-3}	3.7243454×10^{-3}
12	0.0	**0.0**	**0.0**	**0.0**	0.0
13	0.0	1.0×10^{-9}	5.214×10^{-7}	2.41×10^{-8}	7.49×10^{-8}
14	1.0	**0.998**	**0.998**	**0.998**	0.0
15	0.000307	4.3612055×10^{-3}	4.3612227×10^{-3}	4.3612060×10^{-3}	2.5×10^{-9}
16	-1.031628	**-1.031628**	**-1.031628**	**-1.031628**	0.0
17	0.398	**0.397**	**0.397**	**0.397**	0.0
18	3.0	**3.0**	**3.0**	**3.0**	0.0
19	-3.86	**-3.86**	**-3.86**	**-3.86**	0.0
20	-3.32	**-3.32**	**-3.32**	**-3.32**	0.0
21	-10.1422	**-10.15**	**-10.15**	**-10.15**	0.0
22	-10.3909	**-10.40**	**-10.40**	**-10.40**	0.0
23	-10.53	**-10.53**	-5.22	-10.38	0.76919

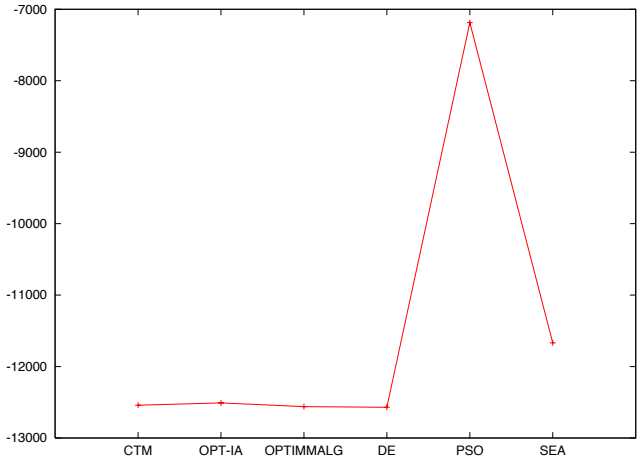
In order to statistically analyze the robustness of the results obtained by CTM, we use the box plot method [26] to visualize the distribution of the 50 values found by CTM for each problem. We discart the problem where CTM found the optimum on each run. In Figures 3a) to 3d) x-axis (Column Number) indicates the functions and y-axis (Values) corresponds to the best objective value found for each run. In Figure 3a) shows the box plot for functions with optimum zero, where 1≡ f_1, 2≡ f_2, 3≡ f_4, 4≡ f_5, 5≡ f_6, 6≡ f_9, 7≡ f_{10}, 8≡ f_{11} and 9≡ f_{13}. Here, we can see CTM is robust except for f_5 and f_9 where the values found for each run are widely dispersed. Figures 3b) to 3d) show the box plots for f_8, f_{15} and f_{23}, respectively. For them, CTM is robust, only found a few exceptional solutions.



a)



b)



c)

Figure 2: a) Best Means obtained by the AIS; b) Best Means obtained by CTM, DE, PSO and SEA; c) Best Means of f_8 obtained obtained for each approach

Table 4: Performance Comparison among CTM, OPT-IMMALG, DE, PSO, SEA and OPT-IA

Function	Optimum	*CTM*	*OPT-IMMALG*	*DE*	*PSO*	*SEA*	*OPT-IA*
1	0.0	7.0×10^{-10}	**0.0**	**0.0**	**0.0**	1.789×10^{-3}	9.23×10^{-12}
2	0.0	1.4591×10^{-6}	**0.0**	**0.0**	**0.0**	2.77×10^{-4}	**0.0**
3	0.0	**0.0**	**0.0**	2.02×10^{-9}	0.0	1.589×10^{-2}	**0.0**
4	0.0	8.6612784×10^{-3}	**0.0**	3.85×10^{-8}	2.107×10^{-16}	1.982×10^{-2}	1.0×10^{-2}
5	0.0	$3.892737636 \times 10^{-1}$	12	**0.0**	4.026	31.3189	3.02
6	0.0	2.0×10^{-2}	**0.0**	**0.0**	4×10^{-2}	**0.0**	0.2
7	0.0	**0.0**	1.521×10^{-5}	4.939×10^{-3}	1.908×10^{-3}	7.106×10^{-4}	3.0×10^{-3}
8	-12569.5	-12540.95	-12560.41	-12569.48	-7187.0	-11669.0	-12508.38
9	0.0	7.8122983770	**0.0**	**0.0**	49.17	0.71789	19.98
10	0.0	6.4684×10^{-6}	**0.0**	-1.19×10^{-15}	1.4	1.0468×10^{-2}	18.98
11	0.0	2.0773141×10^{-3}	**0.0**	**0.0**	2.35×10^{-2}	4.63669×10^{-3}	7.7×10^{-2}
12	0.0	**0.0**	1.77×10^{-21}	**0.0**	3.819×10^{-1}	4.56×10^{-6}	0.137
13	0.0	2.41×10^{-8}	1.686×10^{-21}	-	-	-	1.51
14	1.0	**0.998**	**0.998**	-	-	-	1.02
15	0.000307	4.3612060×10^{-3}	3.2×10^{-4}	-	-	-	7.1×10^{-4}
16	-1.031628	**-1.031628**	-1.013	-	-	-	-1.03158
17	0.398	**0.397**	0.423	-	-	-	**0.398**
18	3.0	**3.0**	5.837	-	-	-	**3.0**
19	-3.86	**-3.86**	-3.72	-	-	-	-3.72
20	-3.32	**-3.32**	-3.3292	-	-	-	-3.31
21	-10.1422	**-10.15**	**-10.153**	-	-	-	-9.11
22	-10.3909	**-10.40**	**-10.402**	-	-	-	-9.86
23	-10.53	-10.38	**-10.536**	-	-	-	-9.96

7 Conclusions and Future Work

This paper presents a novel model AIS for solving global optimization problems. It is called CTM and it is based on the process that suffers the T-cells. The model operates on four populations: Virgins, Effectors (CD4 and CD8) and Memory. The cells in each population have a different representation and the processes they are subject do not are the same.

The approach was found to be competitive in a well-known benchmark commonly adopted in the specialized literature on global optimization problems. The approach was also found to be robust and able to converge to optimum solutions in most cases and very good solutions in others cases. CTM was compared with five different bio-inspired approaches (OPT-IMMALG, OPT-IA, DE, PSO and SEA) and it was very competitive.

We argue that the mutation operators adopted by our approach is capable of performing an efficient local search over each clone, which allows the algorithm to improve the found solutions .

Although there is room for improving our proposed, we have empirically shown that this approach is able of dealing with a variety of global optimization problems (i.e., unimodal functions, multimodal functions with many and a few local optima).

Future work will be dedicated to improve the quality of some solutions found, so that the approach can be competitive with respect to the algorithms representative of the state-of-the-art in the area. For example, we plan to analyze alternative mutation schemes. Besides, we are working on the application of this model in dynamic global optimization problems and constrained optimization problems .

References

[1] Jason Brownlee. Clonal selection algorithms. Technical report id: 070209a, Victoria, Australia: Complex Intelligent Systems Laboratory (CIS), Centre for Information Technology Research (CITR), Faculty of Information and Communication Technologies (ICT), Swinburne University of Technology, 2007.

[2] Kumar Chellapilla. Combining mutation operators in evolutionary programming. IEEE Trans. on Evolutionary Computation, vol. 2, no. 3, pp. 91-96, Sept. 1998.

[3] V. Cutello, G. Morelli, G. Nicosia, and M. Pavone. Immune algorithms with aging operators for the string folding problem and the protein folding problem. In *EvoCOP*, pages 80–90, 2005.

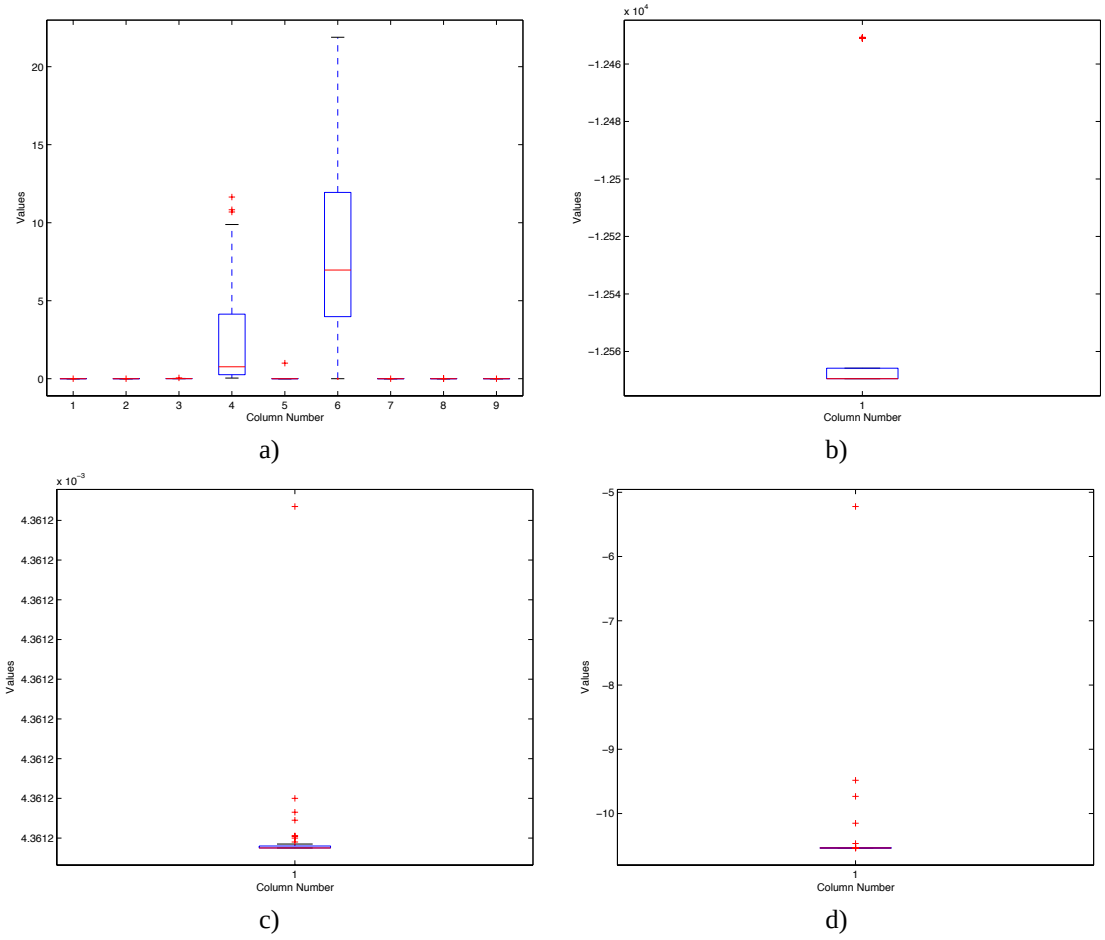


Figure 3: a) Boxplot for f_1, f_2, f_4, f_5, f_6, f_9, f_{10}, f_{11} and f_{13}; b) Boxplot for f_8; c) Boxplot for f_{15}; d) Boxplot for f_{23}

[4] V. Cutello, G. Narzisi, G. Nicosia, and M. Pavone. An immunological algorithm for global numerical optimization. 7th International Conference on Artificial Evolution, EA'05, October 26-28 2005, University of Lille, France, Springer-Verlag, Lecture Notes in Computer Science, vol. 3871, pp. 284-295.

[5] V. Cutello, G. Narzisi, G. Nicosia, and M. Pavone. Clonal selection algorithms: A comparative case study using effective mutation potentials. In *ICARIS*, pages 13–28, 2005.

[6] V. Cutello, G. Narzisi, G. Nicosia, and M. Pavone. An immunological algorithm for global numerical optimization. In *Artificial Evolution*, pages 284–295, 2005.

[7] V. Cutello, G. Narzisi, G. Nicosia, M. Pavone, and G. Sorace. How to escape traps using clonal selection algorithms. In *ICINCO (1)*, pages 322–326, 2004.

[8] V. Cutello and G. Nicosia. An immunological approach to combinatorial optimization problems. In *IBERAMIA 2002: Proceedings of the 8th Ibero-American Conference on AI*, pages 361–370, London, UK, 2002. Springer-Verlag.

[9] V. Cutello and G. Nicosia. Multiple learning using immune algorithms. In *4th International Conference on Recent Advances in Soft Computing, RASC 2002*, pages 102–107, Nottingham, UK, December 2002.

[10] V. Cutello and G. Nicosia. *Recent Developments in Biologically Inspired Computing*, chapter The Clonal Selection Principle for in silico and in vitro Computing. Idea Group Publishing, Hershey, PA, USA, 2004.

[11] V. Cutello, G. Nicosia, and M. Pavone. Real coded clonal selection algorithm for unconstrained global optimization using a hybrid inversely proportional hypermutation operator. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 950–954, New York, NY, USA, 2006. ACM.

[12] D. Dasgupta and F. Nino. *Immunological Computation: Theory and Applications*. Auerbach Publications, Boston, MA, USA, 2008.

[13] Dipankar Dasgupta, Senhua Yu, and Nivedita Sumi Majumdar. MILA - Multilevel Immune Learning Algorithm. In *GECCO*, pages 183–194, 2003.

[14] S. Dasgupta, A. Biswas, S. Das, B. K. Panigrahi, and A. Abraham. A micro-bacterial foraging algorithm for high-dimensional optimization. In *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, pages 785–792, May 2009.

[15] L. N. de Castro and F. J. Von Zuben. The clonal selection algorithm with engineering applications. In *Artificial Immune Systems*, pages 36–39, Las Vegas, Nevada, USA, 8 2000.

[16] L.N. de Castro and F.J. Von Zuben. Learning and optimization using the clonal selection principle. *Evolutionary Computation, IEEE Transactions on*, 6(3):239–251, Jun 2002.

[17] F. Olivetti de França, F. J. Von Zuben, and L. N. de Castro. An artificial immune network for multimodal function optimization on dynamic environments. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 289–296, New York, NY, USA, 2005. ACM.

[18] Maoguo Gong, Licheng Jiao, Lining Zhang, and Wenping Ma. Improved real-valued clonal selection algorithm based on a novel mutation method. In *Intelligent Signal Processing and Communication Systems, 2007. ISPACS 2007. International Symposium on*, pages 662–665, 28 2007-Dec. 1 2007.

[19] Andrew Hone and Johnny Kelsey. Optima, extrema, and artificial immune systems. In *Artificial Immune Systems*, volume 3239/2004 of *Lecture Notes in Computer Science*, pages 80–90. Springer Berlin / Heidelberg, 2004.

[20] N. Jerne. Towards a network theory of the immune system. *Annales d'immunologie*, 125C(1-2):373–389, January 1974.

[21] N. K. Jerne. Clonal selection in a lymphocyte network. *Cellular Selection and Regulation in the Immune Response*, 29:39–48, 1974.

[22] J. Kelsey, J. Timmis, and A. Hone. Chasing chaos. In *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, volume 1, pages 413–419, Dec. 2003.

[23] Johnny Kelsey and Jon Timmis. Immune inspired somatic contiguous hypermutation for function optimisation. In *GECCO*, pages 207–218, 2003.

[24] Li Liu and Wenbo Xu. A cooperative artificial immune network with particle swarm behavior for multimodal function optimization. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 1550–1555, June 2008.

[25] Yidan Luo and Zhongyang Jiang. An improved clonal selection algorithm and its application in function optimization problems. In *Intelligent Information Technology Application, 2008. IITA '08. Second International Symposium on*, volume 2, pages 118–121, Dec. 2008.

[26] Stephan Morgenthaler. Exploratory data analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(1):33–44, July/August 2009. John Wiley & Sons, Inc.

[27] Leandro Nunes de Castro and Jonathan Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer-Verlag, New York, 2002.

[28] J. Timmis and M. Neal. A resource limited artificial immune system for data analysis. *Knowledge-Based Systems*, 14:121–130, 2001.

[29] Jon Timmis, Johnny Kelsey, and C. Edmonds. Assessing the performance of two immune inspired algorithms and a hybrid genetic algorithm for function optmisation. In *In Proceedings of the Congress on Evolutionary Computation (CEC04)*, pages 1044–1051, 2004.

[30] J. Versterstrom and R. Thomsen. A comparative study of differential evolution, particle swarm optimization and evolutionary algorithms on numerical benchmark problems. Congress on Evol. Comp., CEC04, vol. 1, pp. 1980-1987, 2004.

[31] Andrew Watkins and Jon Timmis. Artificial immune recognition system (airs): Revisions and refinements. In J. Timmis and P.J. Bentley, editors, *1st International Conference on Artificial Immune Systems (ICARIS2002)*, pages 173–181, University of Kent at Canterbury, September 2002. University of Kent at Canterbury Printing Unit.

[32] Andrew Watkins and Jon Timmis. Exploiting parallelism inherent in airs, an artificial immune classifier. In G. Nicosia, V. Cutello, P. Bentley, and J. Timmis, editors, *3rd International Conference on Artificial Immune Systems (ICARIS2004)*, number 3239 in Lecture Notes in Computer Science (LNCS), pages 13–16, Catania, Italy, September 2427-438004. University of Kent at Canterbury Printing Unit, Springer.

[33] Andrew B. Watkins. Airs: A resource limited artificial immune classifier. Master's thesis, Mississippi State University, MS. USA., December 2001.

[34] Andrew B. Watkins and Lois C. Boggess. A resource limited artificial immune classifier. In *Congress on Evolutionary Computation, Part of the 2002 IEEE World Congress on Computational Intelligence*, pages 926–931, Honolulu, HI, USA, May 2002.

[35] Jie Yang, Maoguo Gong, Licheng Jiao, and Lining Zhang. Improved clonal selection algorithm based on lamarckian local search technique. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 535–541, June 2008.

[36] Xin Yao, Yong Liu, and Guangming Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3:82–102, 1999.