Artificial Immune System for Solving Global Optimization Problems

Victoria S. Aragón, Susana C. Esquivel *
Universidad Nacional de San Luis
Ejército de los Andes 950
(5700) San Luis - Argentina
{vsaragon, esquivel}@unsl.edu.ar

and

Carlos A. Coello Coello †
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 and we also compare our results with the results obtained by different bio-inspired approaches.

Keywords: Artificial Immune System, Global Optimization Problems.

Resumen

En este trabajo, se presenta un nuevo modelo de Sistema Inmune Artificial (SIA) basado en los procesos que sufren las células T para resolver problemas de optimización global. El modelo, denominado MCT, trabaja sobre cuatro poblaciones con diferentes representaciones para las células y cada población atraviesa por distintos procesos. Se validó el modelo con 23 funciones tomadas de la literatura especializada. El modelo es comparado con diferentes enfoques bio-inspirados.

Palabras claves: Sistema Immune Artificial, Problemas de Optimización Global.

*Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
†CINVESTAV-IPN (Evolutionary Computation Group)

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 [5]. 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.

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.

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 [7].

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 [1].

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 clonal selection principle. OPT-AINET was validated with 18 functions taken from the specialized literature [3].

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 [6].

Cutello V. and Nicosia G. and Pavone M. propose an improved version of OPT-IA [6] 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 [2] .

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 immunitary 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 trigged: 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, 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 representation. A cell from CD4 is composed by:
  - A *TCR* (TCR_b): represented by a bit string, 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 the effector cells implies to select a set of activator (or stimulating) cells and for each of them the cell which be the receptor of the stimulus. Then, the stimulated (or activated) cells proliferate and differentiate, according to their corresponding levels.
- The stimulating cells are those with the lowest recognizing level. The stimulated cells are those with the highest recognizing level.
- At the beginning the proliferation level of each stimulated cell is given by a random value, but then it is determined taking into account the proliferation level of its stimulating cells. If the stimulated cell is better than the stimulating cells, the first mantain its own proliferation level, otherwise the stimulated cells receive a level 10% less than the stimulating cell.
- 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, the best cell from CD4 is include into CD8, replacing the worst cell in CD8.
- Each type of cell has its own differentiation process, which is blind to their representation and population.

Differentiation for CD4 : the number of bits that will be changed is given by the proliferation level, each decision variable and the bit are chosen in a random way and the bit changes according to a mutation (or reaction) probability prob$_{mut-CD4}$.

Differentiation for CD8 : the number of decision variables that will be changed is given by the proliferation level, each variable is chosen in a random way and it changes according to:

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

where x and $x^{'}$ are are the original and mutated decision variables, respectively. $N(0,1)$ refers to a random number with a uniform distribution between (0,1). lu and ll are the upper and lower bounds of x and gen is the current generation 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 $r = \left(\frac{N(0,lu-ll)}{10000000 gen} \right)^{N(0,1)}$, 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 proliferation level, each variable is chosen in a random way and it changes according to:

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

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

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

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 [7] (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

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 predeterminated number of evaluations do
    while Repeat a predeterminated number of times (rep_CD4) do
        Active_CD4();
    end while
    Sort_CD4();
    Comunication_CD4_CD8();
    while Repeat a predeterminated number of times (rep_CD8) do
        Active_CD8();
    end while
    Sort_CD8();
    Insert_CDs_en_MC();
    while Repeat a predeterminated number of times (rep_MC) do
        Active_MC();
    end while
    Sort_CM();
    Statistics();
end while
```

within $[0, 1]$. All problems are minimization problems. 50 independents runs were performed for each problem and the parameter settings are given in Table 2.

Our results are compared with respect to a Differential Evolution algorithm (DE) [4], a Particle Swarm Optimizer (PSO) [4], a simple Evolutionary Algorithm (SEA) [4], an immunological algorithm for continuous global optimization problems (OPT-IA) [6] and an improved version of OPT-IA (OPT-IMMALG) [2]. 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.

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

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
$f_{12}(\mathbf{x}) = \frac{\pi}{n} 10 sin^2(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 sin^2(\pi y_{i+1})] +$ $(y_n - 1)^2 + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$, $y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	30	$[-50, 50]^n$	0
$f_{13}(\mathbf{x}) = 0.1 \left\{sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + sin^2(3\pi x_{i+1})]\right.$ $\left. + (x_n - 1)[1 + sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	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
$f_{18}(\vec{x}) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2$ $+6x_1 x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2$ $-36x_1 x_2 + 27x_2^2)]$	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

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

Table 2: Setting Parameters for each Problem

Function	Evaluations	$\mid VC \mid$	$\mid CD4 \mid$	$\mid CD8 \mid$	$\mid MC \mid$	$\text{prob}_{mut-CD4}$	rep_{CD4}	rep_{CD8}	rep_{MC}
1	150000	100	20	20	20	0.01	800	10	10
2	200000	10	5	5	5	0.01	800	10	10
3	500000	10	5	5	5	0.01	800	10	10
4	500000	10	5	5	5	0.01	800	10	10
5	150000	100	20	20	20	0.01	800	10	10
6	150000	100	20	20	20	0.01	800	10	10
7	300000	100	20	20	20	0.02	800	10	10
8	900000	10	5	5	5	0.01	800	10	10
9	500000	10	5	5	5	0.02	80	10	10
10	150000	10	5	5	5	0.01	800	10	10
11	200000	100	20	20	20	0.02	800	10	10
12	150000	10	5	5	5	0.01	800	10	10
13	150000	100	20	20	20	0.02	800	10	10
14	10000	100	20	20	20	0.02	800	10	10
15	400000	100	20	20	20	0.02	800	10	10
16	10000	100	20	20	20	0.02	800	10	10
17	10000	100	20	20	20	0.02	800	10	10
18	10000	100	20	20	20	0.02	800	10	10
19	10000	100	20	20	20	0.02	800	10	10
20	20000	100	20	20	20	0.02	800	10	10
21	10000	100	20	20	20	0.02	800	10	10
22	10000	100	20	20	20	0.02	800	10	10
23	10000	100	20	20	20	0.01	100	10	10

groups. Figures 1a) and 1b) show the best mean obtained by CTM, OPT-IMMALG and OPT-IA and CTM, DE, PSO and SEA, respectively, for all test functions except f_8. Figure 1c) shows the mean obtained by all approaches.

Table 3: Results obtained by CTM

Function	Optimum	*Best*	*Worst*	*Mean*	*Std.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

Table 4: Performance Comparision 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 .

ACKNOWLEDGMENTS

The first two authors acknowledge support from the Universidad Nacional de San Luis and the ANPCYT through project PICT2005. The third author acknowledges support from the Consejo Nacional de Ciencia y Tecnología (CONACyT) through project number 42435-Y.

REFERENCES

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

[2] Vincenzo Cutello, Giuseppe Nicosia, and Mario 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.

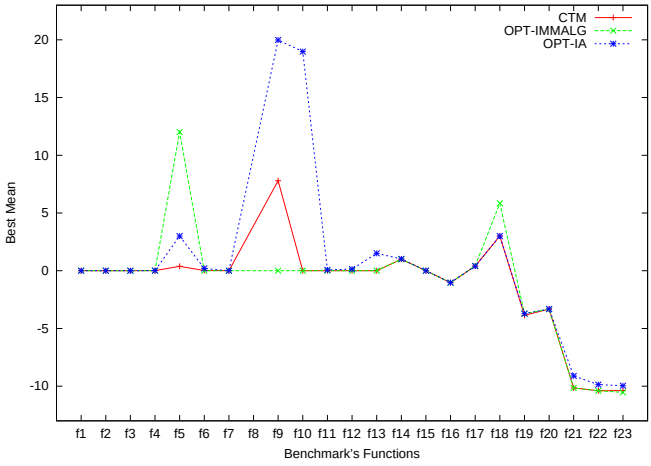
[3] Fabricio Olivetti de França, Fernando J. Von Zuben, and Leandro Nunes de Castro. An artificial immune network for multimodal function optimization on dynamic environments. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 289–296, New York, NY, USA,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

[4] Versterstrom J. and Thomsen R. 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.

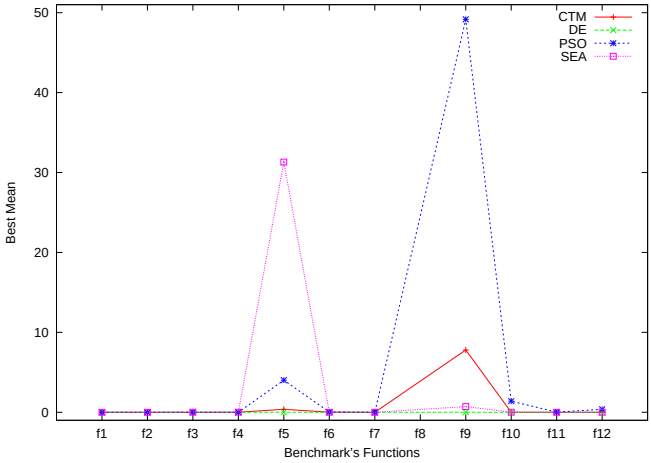
[5] Leandro Nunes de Castro and Jonathan Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer-Verlag, New York, 2002.

[6] Cutello V, Narzisi G, Nicosia G, and Pavone M. 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.

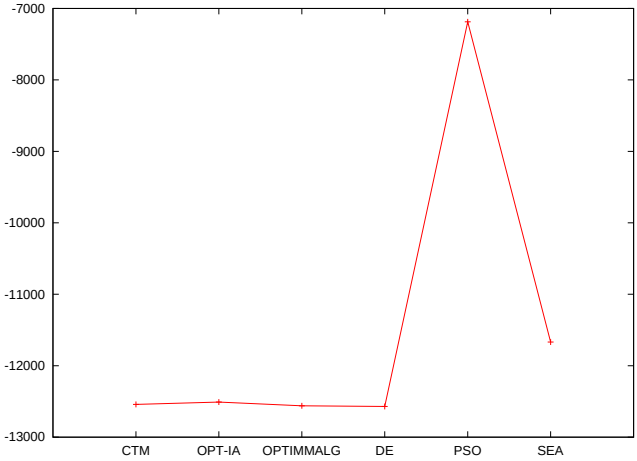
[7] Xin Yao, Yong Liu, and Guangming Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3:82–102, 1999.



a)



b)



c)

Figure 1: 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