An Approach to Solve Multiobjective Optimization Problems Based on an Artificial Immune System

Carlos A. Coello Coello and Nareli Cruz Cortés
CINVESTAV-IPN (Evolutionary Computation Group)
Departamento de Ingeniería Eléctrica/Sección de Computación
Av. IPN No. 2508, Col. San Pedro Zacatenco, México, D. F. 07300
ccoello@cs.cinvestav.mx, nareli@computacion.cs.cinvestav.mx

Abstract

In this paper, we propose an algorithm to solve multiobjective optimization problems (either constrained or unconstrained) using the clonal selection principle. Our approach is compared with respect to another algorithm that is representative of the state-of-the-art in evolutionary multiobjective optimization. For our comparative study, two metrics are adopted and graphical comparisons with respect to the true Pareto front of each problem are also included. Results indicate that the proposed approach is very promising.

1 Introduction

The immune system is one of the most important biological mechanisms humans possess since our own life depends on it. In recent years, several researchers have developed computational models of the immune system that attempt to capture some of their most remarkable features such as its self-organizing capability [11, 9].

From the information processing perspective, the immune system can be seen as a parallel and distributed adaptive system [10, 3]. It is capable of learning, it uses memory and is able of associative retrieval of information in recognition and classification tasks. Particularly, it learns to recognize patterns, it remembers patterns that it has been shown in the past and its global behavior is an emergent property of many local interactions [3]. All these features of the immune system provide, in consequence, great robustness, fault tolerance, dynamism and adaptability [9]. These are the properties of the immune system that mainly attract researchers to try to emulate it in a computer.

In this paper, we propose an approach to solve multiobjective optimization problems (either with or without constraints) based on the clonal selection principle.

2 The Immune System

The main goal of the immune system is to protect the human body from the attack of foreign (harmful) organisms. The immune system is capable of distinguishing between the normal components of our organism and the foreign material that can cause us harm (e.g., bacteria). These foreign organisms are called *antigens*.

The molecules called *antibodies* play the main role on the immune system response. The immune response is specific to a certain foreign organism (antigen). When an antigen is detected, those antibodies that best recognize an antigen will proliferate by cloning. This proccess is called *clonal selection principle* [4].

The new cloned cells undergo high rate mutations or *hypermutation* in order to increase their receptor population (called repertoire). These mutations experienced by the clones are proportional to their affinity to the antigen.

The highest affinity antibodies experiment the lowest mutation rates, whereas the lowest affinity antibodies have high mutation rates. After this mutation process ends, some clones could be dangerous for the body and should therefore be eliminated.

After these clonation and hypermutation processes finish, the immune system has improved the antibodies' affinity, which results on the antigen neutralization and elimination.

At this point, the immune system must return to its normal conditions, eliminating the excedent cells. However, some cells remain circulating throughout the body as memory cells. When the immune system is later attacked by the same type of antigen (or a similar one), these memory cells are activated, presenting a better and more efficient response. This second encounter with the same antigen is called *secondary response*.

The algorithm proposed in this paper is based on the clonal selection principle previously described.

3 Multiobjective Optimization

Multiobjective optimization (also called multicriteria optimization, multiperformance or vector optimization) can be defined as the problem of finding [15]:

> a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term "optimize" means finding such a solution which would give the values of all the objective functions acceptable to the designer.

Formally, we can state the general multiobjective optimization problem (MOP) as follows:

Definition 1 (General MOP): *Find the vector* $\vec{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ *which will satisfy the* m *inequality constraints:*

$$g_i(\vec{x}) \geq 0 \quad i = 1, 2, \ldots, m \tag{1}$$

the p *equality constraints*

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \ldots, p \tag{2}$$

and optimizes the vector function

$$\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})]^T \tag{3}$$

where $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ *is the vector of decision variables.* □

In other words, we wish to determine from among the set $\mathcal{F}$ of all numbers which satisfy (1) and (2) the particular set $x_1^*, x_2^*, \ldots, x_n^*$ which yields the optimum values of all the k objective functions of the problem.

Another important concept is that of Pareto optimality, which was stated by Vilfredo Pareto in the XIX century [16], and constitutes by itself the origin of research in multiobjective optimization:

Definition 2 (Pareto Optimality:): *We say that* $\vec{x}^* \in \mathcal{F}$, *is* **Pareto optimal** *if for every* $\vec{x} \in \Omega$ *and* $I = \{1, 2, \ldots, k\}$ *either,*

$$\bigwedge_{i \in I} (f_i(\vec{x}) = f_i(\vec{x}^*)) \tag{4}$$

or, there is at least one $i \in I$ *such that (assuming maximization)*

$$f_i(\vec{x}) \leq f_i(\vec{x}^*) \tag{5}$$

□

In words, this definition says that $\vec{x}^*$ is Pareto optimal if there exists no feasible vector $\vec{x}$ which would increase some criterion without causing a simultaneous decrement in at least one other criterion.

Pareto optimal solutions are also termed non-inferior, admissible, or efficient solutions [2]; their corresponding vectors are termed nondominated. These solutions may have no clearly apparent relationship besides their membership in the Pareto optimal set. This is the set of all solutions whose corresponding vectors are nondominated with respect to all other comparison vectors. When plotted in objective space, the nondominated vectors are collectively known as the Pareto front.

4 The Proposed Approach

As indicated before, our algorithm is based on the *clonal selection principle*, modeling the fact that only the highest affinity antibodies to the antigens will proliferate. Our algorithm uses the concept of Pareto dominance to generate nondominated vectors. Also, an external (or secondary) memory is used to store nondominated vectors found along the evolutionary process, in order to move towards the true Pareto front over time (this can be seen as a form of elitism in evolutionary multiobjective optimization [2]).

4.1 The Algorithm

Our algorithm is the following:

1. Generate randomly the initial population.

2. Initialize the secondary memory so that it is empty.

3. Determine for each individual in the population, if it is (Pareto) dominated or not. For constrained problems, determine if an individual is feasible or not.

4. Split the population into antigens and antibodies. The division criterion is Pareto dominance (i.e., nondominated individuals are the antigens and dominated individuals are the antibodies). In constrained problems, feasible individuals are antigens, too. Note that either of the two criteria (Pareto dominance or feasibility) is sufficient for an individual to be considered an antigen. However, to guide the search properly, we distinguish between "very good" (or ideal) antigens and those which are only "good". For that sake, we assign a weight (w) to each antigen according to the following rules:

   - $w = 4$ for nondominated and feasible antigens (the best ones).
   - $w = 3$ for nondominated antigens (even if infeasible).
   - $w = 2$ for feasible antigens (even if they are dominated).

   Note that in the previous rules, Pareto dominance is given more importance than feasibility. These values were arbitrarily adopted to give more or less importance to each of the cases previously indicated. Note however, that the same values are adopted in all the examples presented in this paper. Also, note that in unconstrained problems, all nondominated individuals are made antigens with a $w = 2$.

5. Copy the antigens (with $w = 4$ for constrained problems and with $w = 2$ for unconstrained problems) to the secondary memory.

6. Select an antigen (regardless of its weight) at random.

7. Assign a fitness value to each of the antibodies according to their matching value (Z) with respect to the antigen (randomly) chosen from the previous step (see Figure 1). Note that a new antigen is randomly selected for each antibody.

8. Select the Q fittest antibodies from the antibodies pool where the fitness criterion is defined by the value of Z.

9. Create a number N of copies of the antibodies selected.

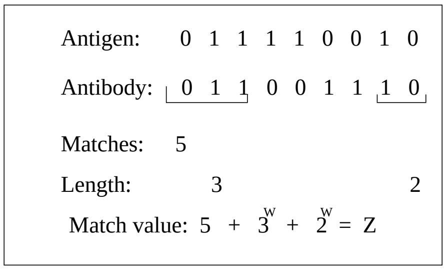

Figure 1: Matching measure between an antigen and an antibody. The weights w are used to increase the value of Z when an antibody matches a highly desirable antigen (i.e., nondominated and feasible).

10. Assign a mutation rate (MR) to each clone, according to their similarity with an antigen randomly chosen. The higher the similarity the lower the mutation rate, and viceversa.

11. Apply mutation rate MR to each clone.

12. The new population is formed by the union of the original antibodies and their clones.

13. The population size is returned to its original value, allowing the nondominated individuals (and the feasible ones if dealing with a constrained problem) survive.

14. Go back to step 3 until convergence occurs or after reaching a certain (predetermined) number of iterations.

The antigen-antibody matching measure (Z) adopted in this paper is adapted from Farmer's proposal [7]. This matching measure counts the number of matching bits of the two strings compared as well as the number of consecutive matching bits. For example, if we have three contiguous similarities on the strings we add a value of 3 raised to its w value to the total matching measure (see figure 1).

Note that this algorithm is not really a genetic algorithm since no sexual recombination takes place. Instead, only a clonation of individuals is used to generate the new population of the algorithm.

4.2 Secondary Memory

We use a secondary or external memory as an elitist mechanism in order to maintain the best solutions found along the process. The individuals stored in this memory are all nondominated not only with respect to each other but also with respect to all of the previous individuals who attempted to enter the external

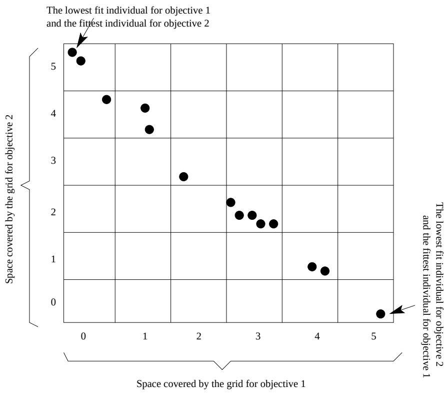

Figure 2: An adaptive grid to hadle the secondary memory

memory. Therefore, the external memory stores our approximation to the true Pareto front of the problem.

In order to enforce a uniform distribution of nondominated solutions that cover the entire Pareto front of a problem, we use the adaptive grid proposed by Knowles and Corne [13] (see Figure 2).

Ideally, the size of the external memory should be infinite. However, since this is not possible in practice, we must set a limit to the number of nondominated solutions that we want to store in this secondary memory. By enforcing this limit, our external memory will get full at some point even if there are more nondominated individuals wishing to enter. When this happens, we use an additional criterion to allow a nondominated individual to enter the external memory: region density (i.e., individuals belonging to less densely populated regions are given preference).

The algorithm for the implementation of the adaptive grid is the following:

1. Divide objective function space according to the number of subdivisions set by the user.
2. For each individual in the external memory, determine the cell to which it belongs.
3. If the external memory is full, then determine which is the most crowded cell.
4. To determine if a certain antigen is allowed to enter the external memory, do the following:
   - If it belongs to the most crowded cell, then it is not allowed to enter.
   - Otherwise, the individual is allowed to enter. For that sake, we eliminate a (randomly chosen) individual that belongs to the most crowded cell in order to have an available slot for the antigen.

5 Experiments

In order to validate our approach, we used several test functions reported in the standard evolutionary multiobjective optimization literature [5, 20, 2]. In each case, we generated the true Pareto front of the problem (i.e., the solution that we wished to achieve) by enumeration using parallel processing techniques. Then, we plotted the Pareto front generated by our algorithm, which we call the multiobjective immune system algorithm (MISA). The results indicated below were found using the following parameters: Maximum number of iterations = 150, population size = 70, clonation rate = 0.8, number of clones = 15, size of the external memory = 100. The above parameters produce a total of 138,000 fitness function evaluations.

MISA was compared against the micro-genetic algorithm for multiobjective optimization, which was recently proposed [1]. This algorithm is representative of the state-of-the-art in evolutionary multiobjective optimization and has been found to produce similar or better results than the NSGA-II [6] and PAES [13].

To allow a fair comparison, the micro-GA performed the same number of fitness function evaluations as MISA.

Despite the graphical comparisons performed, the two following metrics were adopted to compare our results:

- **Two Set Coverage (SC)**: This metric was proposed in [22], and it can be termed *relative coverage comparison of two sets*. Consider $X', X'' \subseteq X'$ as two sets of phenotype decision vectors. SC is defined as the mapping of the order pair (X', X'') to the interval $[0, 1]$.

$$SC(X', X'') \triangleq \frac{|\{a'' \epsilon X''; \exists a' \epsilon X' : a' \succeq a''\}|}{|X''|} \quad (6)$$

  If all points in X' dominate or are equal to all points in X'', then by definition $SC = 1$. $SC = 0$ implies the opposite. In general, $SC(X', X'')$ and $SC(X'', X')$ both have to be considered due to set intersections not being empty. Of course, this metric can be used for both spaces (objective function or decision variable space), but in this case we applied it in objective function space. The advantage of this metric is that it is easy to calculate and provides a relative comparison based upon dominance numbers between generations or algorithms.

- **Spacing (S)**: This metric was proposed by Schott [18] as a way of measuring the range (distance) variance of neighboring vectors in the Pareto front known. This metric is defined as:

$$S \triangleq \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\overline{d}-d_i)^2}\ , \tag{7}$$

where $d_i = \min_j(\mid f_1^i(\vec{x}) - f_1^j(\vec{x}) \mid + \mid f_2^i(\vec{x}) - f_2^j(\vec{x}) \mid)$, $i, j = 1, \ldots, n$, $\overline{d}$ is the mean of all d_i, and n is the number of vectors in the Pareto front found by the algorithm being evaluated. A value of zero for this metric indicates all the nondominated solutions found are equidistantly spaced.

The parameters used by the micro-GA for the experiments reported below are the following: maximum number of generations = 8400, population size = 4, number of grid subdivisions = 25, memory size = 50, crossover rate = 0.8, number of iterations to achieve nominal convergence = 4, size of the external memory = 100. We the previous parameters, the micro-GA performs a total of 138,000 fitness function evaluations.

Example 1

Minimize: $F = (f_1(x,y), f_2(x,y))$, where

$$\begin{aligned} f_1(x,y) &= x, \\ f_2(x,y) &= (1+10y) * \\ & \quad [1 - (\frac{x}{1+10y})^\alpha - \frac{x}{1+10y}\sin(2\pi q x)] \end{aligned}$$

and $0 \leq x, y \leq 1$, $q = 4$, $\alpha = 2$.

The comparison of results between the true Pareto front of this example and the Pareto front produced by MISA is shown in Figure 3. Note that the Pareto front is disconnected (it consists of four Pareto curves). In this case: $SC(MISA, micro-GA) = 0.304$ and $SC(micro-GA, MISA) = 0.29$. This indicates a very similar behavior from both algorithms and we can say that there is a tie among the final nondominated solutions produced by the two algorithms. In terms of spacing, the results are presented in Table 1. Note that the average results of MISA are better than those of the micro-GA.

Table 1: Spacing for example 1

	best	average	worst	std.dev.
MISA	0.008853	0.114692	0.62904	0.175955
micro-GA	0.007773	0.177104	0.991838	0.319061

Table 2: Spacing for example 2

	best	average	worst	std.dev.
MISA	0.008853	0.107427	0.209062	0.054843
micro-GA	0.04119	0.1446	1.197458	0.253813

Example 2

Our second example is a two-objective optimization problem proposed by Schaffer [17] that has been used by several researchers [19]:

$$\text{Minimize } f_1(x) = \begin{cases} -x & \text{if } x \leq 1 \\ -2+x & \text{if } 1 < x \leq 3 \\ 4-x & \text{if } 3 < x \leq 4 \\ -4+x & \text{if } x > 4 \end{cases} \tag{8}$$

$$\text{Minimize } f_2(x) = (x-5)^2 \tag{9}$$

and $-5 \leq x \leq 10$.

The comparison of results between the true Pareto front of this example and the Pareto front produced by MISA is shown in Figure 4. In this case: $SC(MISA, micro-GA) = 0.487$ and $SC(micro-GA, MISA) = 0.56$. As in the previous example, these values indicate a very similar behavior from both algorithms and we can say that there is a tie among the final nondominated solutions produced by the two algorithms. In terms of spacing, the results are shown in Table 2. Note again that the average results of MISA are better than those of the micro-GA.

Example 3

The third example is the three-objective function problem proposed by Viennet [21]:

Minimize: $F = (f_1(x,y), f_2(x,y), f_3(x,y))$

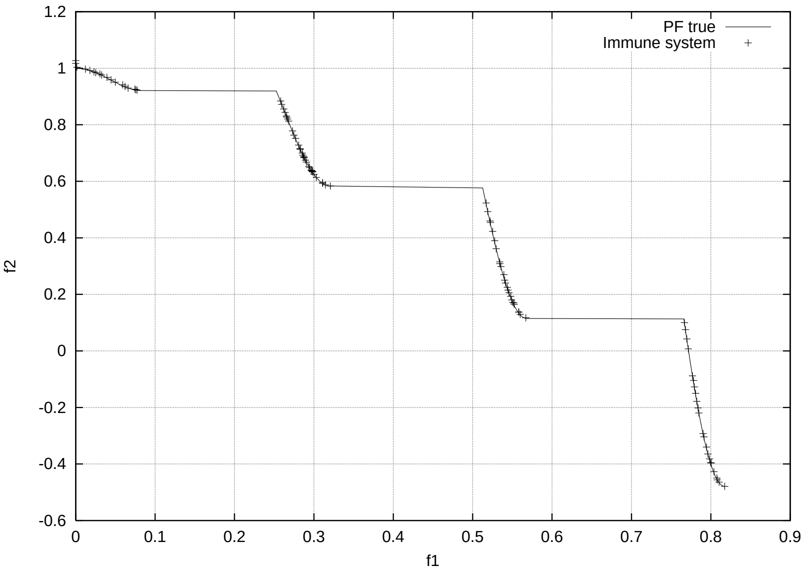

Figure 3: Comparison of results for the first example. The true Pareto front is shown as a continuous line (note that the horizontal segments are NOT part of the Pareto front and are shown only to facilitate drawing the front) and the Pareto front found by MISA is shown as crosses.

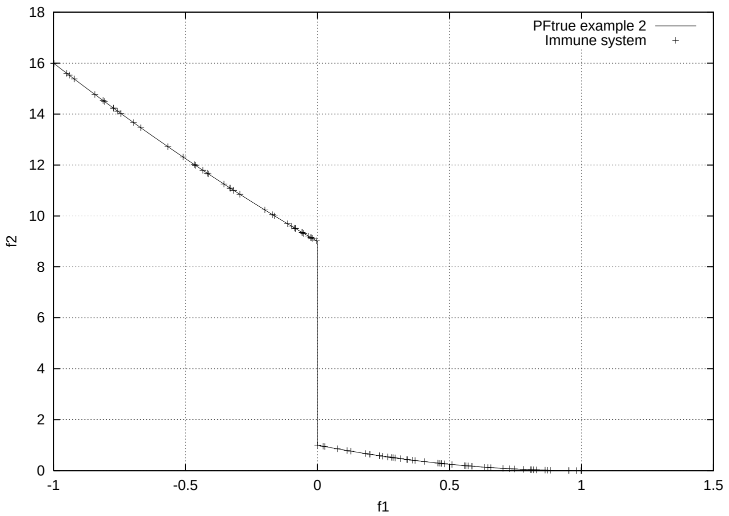

Figure 4: Comparison of results for the second test function. The true Pareto front of the problem is shown as a continuous line (note that the vertical segment is NOT part of the Pareto front and is shown only to facilitate drawing the front) and the Pareto front found by MISA is shown as crosses.

where:

$$
\begin{aligned}
f_1(x,y) &= \frac{(x-2)^2}{2} + \frac{(y+1)^2}{13} + 3, \\
f_2(x,y) &= \frac{(x+y-3)^2}{175} + \frac{(2y-x)^2}{17} - 13, \\
f_3(x,y) &= \frac{(3x-2y+4)^2}{8} + \frac{(x-y+1)^2}{27} \\
&\quad +15
\end{aligned}
$$

and: $-4 \leq x, y \leq 4$, $y < -4x + 4$, $x > -1$, $y > x - 2$.

The comparison of results between the true Pareto front of this example and the Pareto front produced by MISA is shown in Figure 5. In this case: $SC(MISA, micro-GA) = 0.673$ and $SC(micro-GA, MISA) = 0.605$. As in the previous example, these values indicate a very similar behavior from both algorithms and we can say that there is a tie among the final nondominated solutions produced by the two algorithms. In terms of spacing, the results are shown in Table 3. Note that the average results of the micro-GA are better than those of MISA. In this case, MISA had a poorer performance in terms of uniform distribution than the micro-GA.

Example 4

The fourth example was proposed by Kita [12]:

Maximize $F = (f_1(x,y), f_2(x,y))$

where:

$$
\begin{aligned}
f_1(x,y) &= -x^2 + y, \\
f_2(x,y) &= \frac{1}{2}x + y + 1
\end{aligned}
$$

$x, y \geq 0$, $0 \geq \frac{1}{6}x + y - \frac{13}{2}$, $0 \geq \frac{1}{2}x + y - \frac{15}{2}$, $0 \geq 5x + y - 30$.

The comparison of results between the true Pareto front of this example and the Pareto front produced by MISA is shown in Figure 6. In this case: $SC(MISA, micro-GA) = 1.00$ and $SC(micro-GA, MISA) = 0.145$. In this case, MISA produced solutions that clearly dominated or were equal to those generated by the micro-GA (therefore the value of 1.0). This clearly indicates a better behavior of MISA. In terms of spacing, the results are shown in Table 4. In terms of this metric, the average results of the micro-GA are better than those of MISA. Note however, that since the solutions generated by the micro-GA are covered (i.e., dominated) by those produced by MISA, the fact that these solutions have a more uniform distribution is less relevant, since these solutions are poorer than those generated by MISA.

Table 4: Spacing for example 4

	best	average	worst	std.dev.
MISA	0.141532	0.518706	1.145541	0.349627
micro-GA	0.039568	0.115826	0.830159	0.180039

Example 5

Our fifth example is a two-objective optimization problem defined by Kursawe [14]:

$$
\text{Minimize } f_1(\vec{x}) = \sum_{i=1}^{n-1} \left(-10 \exp \left(-0.2 \sqrt{x_i^2 + x_{i+1}^2} \right) \right) \tag{10}
$$

$$
\text{Minimize } f_2(\vec{x}) = \sum_{i=1}^{n} \left(|x_i|^{0.8} + 5 \sin(x_i)^3 \right) \tag{11}
$$

where:

$$
-5 \leq x_1, x_2, x_3 \leq 5 \tag{12}
$$

The comparison of results between the true Pareto front of this example and the Pareto front produced by MISA is shown in Figure 7. In this case: $SC(MISA, micro-GA) = 0.3490$ and $SC(micro-GA, MISA) = 0.96$. In this case, the micro-GA produced solutions that clearly dominated or were equal to those generated by MISA (therefore the value very close to 1.0). This clearly indicates a better behavior of the micro-GA. In terms of spacing, the results are shown in Table 5. Note that the average results of MISA are better than those of the micro-GA. Note however, that since the solutions generated by MISA are covered (i.e., dominated) by those produced by the micro-GA, the fact that these solutions have a more uniform distribution is less relevant, since these solutions are poorer than those generated by the micro-GA.

Summarizing, we can see that our approach has a very competitive behavior with respect to the micro-GA when dealing with unconstrained test functions. However, in constrained test functions is not as competitive (in general), but the results are still acceptable as

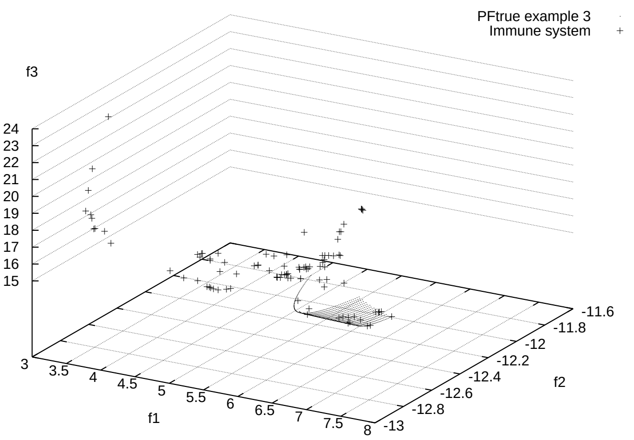

Figure 5: Comparison of results for the third test function. The true Pareto front of the problem is shown as dots and the Pareto front found by MISA is shown as crosses.

Table 3: Spacing for example 3

	best	average	worst	std.dev.
MISA	0.382708	0.515023	0.632426	0.057835
micro-GA	0.270519	0.294236	0.315999	0.012565

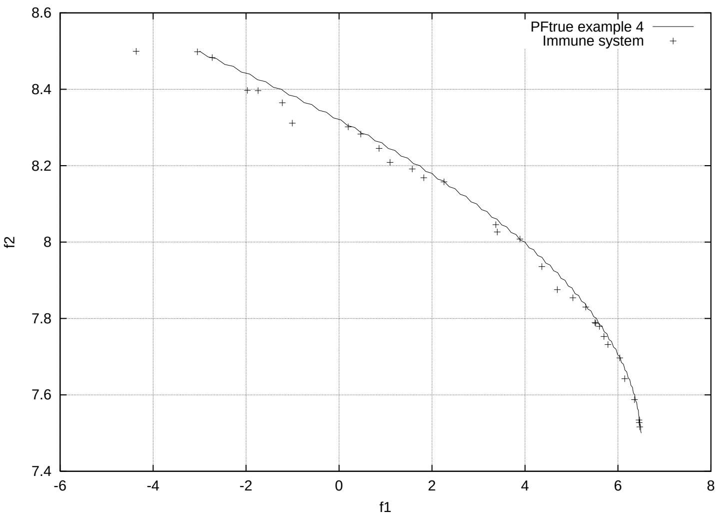

Figure 6: Comparison of results for the fourth test function. The true Pareto front of the problem is shown as a continuous line and the Pareto front found by MISA is shown as crosses.

Table 5: Spacing for example 5

	best	average	worst	std.dev.
MISA	2.008484	2.382588	3.201155	0.292547
micro-GA	2.945237	3.299231	3.905389	0.353365

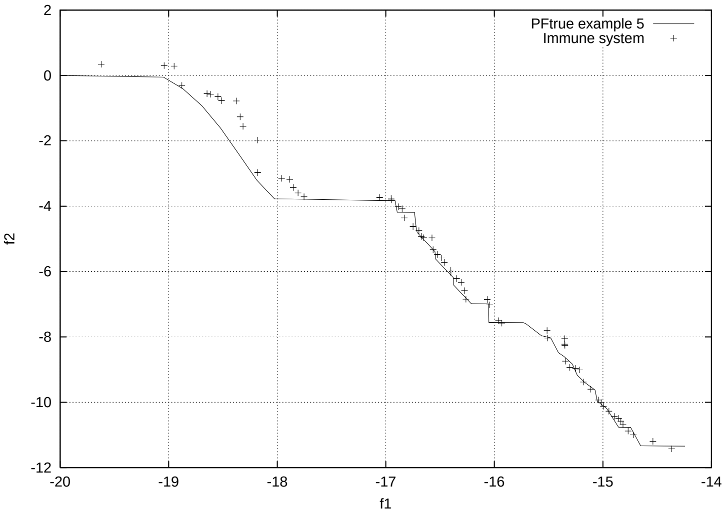

Figure 7: Comparison of results for the fifth test function. The true Pareto front of the problem is shown as a continuous line (note that the horizontal segment is NOT part of the Pareto front and is shown only to facilitate drawing the front) and the Pareto front found by MISA is shown as crosses.

can be seen in the corresponding graphs. Nevertheless, further improvements are required so that MISA can incorporate constraints more efficiently into its fitness function.

6 Conclusions and Future Work

We have presented a new multiobjective optimization algorithm based on the *clonal selection principle*. The approach seems promising and is able to produce results similar or better than those generated by an algorithm that represents the state-of-the-art in evolutionary multiobjective optimization when dealing with unconstrained test functions. However, the algorithm still requires further improvements so that it can handle constraints more efficiently. Such work is currently under way.

Additionally, we will be performing direct comparisons with other evolutionary multiobjective optimization techniques such as PAES [13], the NSGA-II [6] and MOGA [8] with elitism. In such comparative study, additional metrics will be implemented.

Our goal is to produce a highly competitive algorithm (based on the artificial immune system) that represents a viable alternative to solve multiobjective optimization problems of any kind (either constrained or unconstrained).

Acknowledgements

We thank the comments of the anonymous reviewers that greatly helped us to improve the contents of this paper. The first author gratefully acknowledges support from CONACyT through project 34201-A. The second author acknowledges support from CONACyT through a scholarship to pursue graduate studies at the Computer Science Section of the Electrical Engineering Department at CINVESTAV-IPN.

References

[1] Carlos A. Coello Coello and Gregorio Toscano Pulido. Multiobjective Optimization using a Micro-Genetic Algorithm. In Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 274–282, San Francisco, California, 2001. Morgan Kaufmann Publishers.

[2] Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, May 2002. ISBN 0-3064-6762-3.

[3] Dipankar Dasgupta, editor. *Artificial Immune Systems and Their Applications*. Springer-Verlag, Berlin, 1999.

[4] Leandro Nunes de Castro and Fernando José Von Zuben. Artificial Immune Systems: Part I - Basic Theory and Applications. Technical Report TR-DCA 01/99, FEEC/UNICAMP, Brazil, December 1999.

[5] Kalyanmoy Deb. Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation*, 7(3):205–230, Fall 1999.

[6] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[7] J. D. Farmer, N. H. Packard, and A. S. Perelson. The Immune System, Adaptation, and Machine Learning. *Physica D*, 22:187–204, 1986.

[8] Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. Morgan Kauffman Publishers.

[9] Stephanie Forrest and Steven A. Hofmeyr. Immunology as Information Processing. In L.A. Segel and I. Cohen, editors, *Design Principles for the Immune System and Other Distributed Autonomous Systems*, Santa Fe Institute Studies in the Sciences of Complexity, pages 361–387. Oxford University Press, 2000.

[10] Steven A. Frank. *The Design of Natural and Artificial Adaptive Systems.* Academic Press, New York, 1996.

[11] John E. Hunt and Denise E. Cooke. An adaptative, distributed learning systems based on the immune system. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernatics*, pages 2494–2499, 1995.

[12] Hajime Kita, Yasuyuki Yabumoto, Naoki Mori, and Yoshikazu Nishikawa. Multi-Objective Optimization by Means of the Thermodynamical Genetic Algorithm. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature—PPSN IV*, Lecture Notes in Computer Science, pages 504–512, Berlin, Germany, September 1996. Springer-Verlag.

[13] Joshua D. Knowles and David W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

[14] Frank Kursawe. A Variant of Evolution Strategies for Vector Optimization. In H. P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature. 1st Workshop, PPSN I*, volume 496 of *Lecture Notes in Computer Science*, pages 193–197, Berlin, Germany, oct 1991. Springer-Verlag.

[15] Andrzej Osyczka. Multicriteria optimization for engineering design. In John S. Gero, editor, *Design Optimization*, pages 193–227. Academic Press, 1985.

[16] Vilfredo Pareto. *Cours D'Economie Politique*, volume I and II. F. Rouge, Lausanne, 1896.

[17] J. David Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms.* PhD thesis, Vanderbilt University, 1984.

[18] Jason R. Schott. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1995.

[19] N. Srinivas and Kalyanmoy Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.

[20] David A. Van Veldhuizen and Gary B. Lamont. MOEA Test Suite Generation, Design & Use. In Annie S. Wu, editor, *Proceedings of the 1999 Genetic and Evolutionary Computation Conference. Workshop Program*, pages 113–114, Orlando, Florida, July 1999.

[21] Rémy Viennet, Christian Fontiex, and Ivan Marc. New Multicriteria Optimization Method Based on the Use of a Diploid Genetic Algorithm: Example of an Industrial Problem. In J. M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, editors, *Proceedings of Artificial Evolution (European Conference, selected papers)*, pages 120–127, Brest, France, September 1995. Springer-Verlag.

[22] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.