Using Artificial Immune Systems to Solve Optimization Problems

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

Abstract

This paper summarizes a research project focused on the analysis and development of new algorithms based on artificial immune systems to solve optimization problems. Two main types of problems are of our main interest: (1) how to incorporate constraints of any type (linear, nonlinear, equality and inequality) into a genetic algorithm used for single-objective optimization, and (2) how to generate the Pareto optimal set of a multiobjective optimization problem. Our current results indicate that artificial immune systems are a viable alternative both for single- and for multi-objective optimization.

1 Introduction

Artificial immune systems (AIS) are a new research area that takes ideas from our biological immune system to solve complex problems, mainly in engineering and the sciences [11, 5]. The success of our immune system to defend us from diseases is due to its several information processing features that include the following: pattern recognition capabilities, memory, ability to learn, robustness, fault tolerance, and self-organizing capabilities, among others. These are the properties of the immune system that mainly attract researchers to try to emulate it in a computer. Even when the immune system is a very complex system (comparable only to our brain) and its behavior is not fully understood, several computational models of it have been developed so far [11, 5]. Our research has focused on the use of artificial immune systems to solve numerical optimization problems of two types: single-objective and multi-objective, both with and without constraints.

2 Artificial immune system

The aim of the immune system is to keep an organism healthy, recognizing foreign agents (called *antigens*) and defending the organism from them. The main actors in the immune response are the molecules called *antibodies*, whose recognition capabilities are impressive. Antibodies are capable of recognizing an enormous quantity of diverse antigens, from only a fairly limited variety. The information necessary to carry out this recognition task is distributed throughout the genome, and the antibodies must combine segments of such information to form the immune response. When an antigen has been recognized, the antibodies with the highest affinity to it will be cloned creating multiple copies of them. Furthermore, these new cloned molecules will suffer a high rate mutation (or hypermutation), in order to improve their affinity. The mutation rate is proportional to the antibody-antigen affinity. To a higher affinity, a lower mutation rate is used, and viceversa. At this point, the immune system is able to neutralize and eliminate the antigens. Some high affinity clones will remain circulating through the body as memory cells, then at a secondary exposition to the same antige (or a similar one), the immune system will use these memory cells for providing a more efficient and faster response (called secondary response). The clonation and hypermutation processes are called the *clonal selection principle* [11], and this is precisely one of the models adopted in the research reported in this paper.

3 Constraint-Handling with an AIS

Our first interest is to solve single-objective nonlinear optimization problems of the form:

$$\text{Minimize } f(\vec{x}) \tag{1}$$

subject to:

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

$$h_j(\vec{x}) = 0, \quad j = 1, \ldots, p \tag{3}$$

where $\vec{x}$ is the vector of decision variables, n is the number of inequality constraints and p is the number of equality constraints. For solving this problem, we proposed an extension of the algorithm of Hajela & Lee [7] that emulates the invaders recognition process by combining antibodies' libraries in order to attain antigen specificity. The algorithm is based on the mechanism by which the immune system combines gene segments to learn the correct antibodies for the specific antigen. The search process of our approach is led by a genetic algorithm in which the feasible designs will be considered as antibodies and the infeasible designs as antigens. Then, we run a simulation in which the goal is that infeasible designs are "similar" (where similarity is measured in terms of Hamming distances in genotypic space) to feasible ones. The idea is to push infeasible individuals to the feasible region. This approach uses a GA embedded into another one. The outer GA is on charge of optimizing the original objective function, and the inner GA only handles the constraints of the problem. Note however that the computational complexity of the approach is not really (N^2), because the internal genetic algorithm does not evaluate the original fitness function of the problem, but only measures Hamming distances (see [1] for details).

3.1 Use of Parallelism

In order to study the effect of parallelism in our constraint-handling approach, we proposed a *multiple-population GA* with a distributed memory. The population was divided in as many demes (subpopulations) as processors were available. Each deme evolved independently from the others exchanging individuals at regular intervals. Some of the main considerations to take into consideration in the design of our parallel GA were the following:

- The number of demes was equal to the number of available processors and the population was distributed throughout the demes.
- The number of iterations for the AIS simulation (inner GA) was reduced proportionally to the number of demes used.
- For each deme, we applied different genetic operators in order to explore different regions of the search space.

	KM	Serial	2 P SP=5.11	3 P SP=10.71	4 P SP=20.17
	$f(x)$	$f(x)$	$f(x)$	$f(x)$	$f(x)$
AV	-14.71	-14.53	-14.78	-14.59	-14.82
BS	-14.79	-14.78	-14.99	-14.99	-14.99
WS	-14.62	-13.84	-12.99	-11.88	-12.99
SD	N/A	0.23	0.51	0.88	0.4830

Table 1: Comparison of results for the first example. P indicates the number of processors used and SP refers to the speedup achieved. N/A = Not Available. AV = Mean, BS = Best, WS = Worst, SD = Standard Deviation.

The efficiency of a parallel algorithm tends to be measured in terms of its correctness and its speedup. The speedup (SP) of an algorithm is obtained by dividing the processing time of the best serial algorithm (T_s) by the processing time of the parallel version (T_p)[1]: $SP = \frac{T_s}{T_p}$. To obtain the best serial algorithm, we performed a set of experiments to determine the minimum parameters required by our algorithm to operate reasonably well [12]. We compared the results obtained from both the serial and the parallel versions of our algorithm against Koziel & Michalewickz's technique [9], which is one of the best constraint-handling techniques known at date, using the well-known Michalewicz's test function benchmark [10].

Table 1 shows results for one of the test functions used. Note that our serial implementation generates practically the same results that Koziel and Michalewicz's technique (KM), despite the lower computational cost of our approach (150,000 fitness function evaluations vs. 1,400,000 of KM). The parallel versions maintain similar results but with a lower standard deviation. The speedup achieved in this case is remarkable.

4 Multiobjective Optimization

Our second goal was to be able to solve multiobjective optimization problems (MOPs) which are defined as follows:

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

$$g_i(\vec{x}) \geq 0 \quad i = 1, 2, \ldots, m \tag{4}$$

the p equality constraints

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \ldots, p \tag{5}$$

and will optimize the vector function

$$\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})]^T \tag{6}$$

[1] This expression assumes identical processors and identical input sizes (i.e., number of fitness function evaluations in our case).

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

Having several objective functions, the notion of "optimum" changes, because in MOPs, the aim is to find good compromises (or "trade-offs") rather than a single solution as in global optimization. The notion of "optimum" that is most commonly adopted is called *Pareto optimum*. In words, the definition of Pareto optimum says that $\vec{x}^*$ is Pareto optimal if there exists no feasible vector $\vec{x}$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion. The phrase "Pareto optimal" is considered to mean with respect to the entire decision variable space unless otherwise specified. Pareto optimal solutions are also termed *non-inferior*, *admissible*, or *efficient* solutions; their corresponding vectors are termed *nondominated*. We found several difficulties attempting to extend our previous algorithm to deal with MOPs. Basically, the main difficulty was related to the use of similarity measures both for identifying nondominated solutions and maintaining diversity. This led us to use a different approach, based on the *clonal selection principle*, which we briefly described before. In our proposed approach, the set of candidate solutions is represented by binary strings, and are classified as antigens and antibodies. In a first attempt, our antigens were the nondominated individuals, and a feasibility criterion was also adopted if the MOP was constrained. In contrast, antibodies represented individuals that were dominated or feasible. A weight (w) was assigned to each antigen indicating if it was a "very good" or just a "good" antigen ("very good" denoted nondominated and feasible antigens, whereas "good" indicated nondominated solutions (even if infeasible)). These weights were used to affect the antigen-antibody affinity (genotypic matching), promoting both nondominance and feasibility. Then, high affinity antibodies were cloned. These new clones were mutated with rates proportional to their affinity measure. The highest the affinity, the lower the mutation rate used and viceversa. At this point, the population was formed by the set of antigens, antibodies and mutated clones. The population size then was reset to its original value, allowing only the survival of the nondominated individuals. Our approach used a secondary population (an external file) in order to help the algorithm to produce a uniform distribution of nondominated vectors along the Pareto front. For that sake, we adopted the adaptive grid proposed by Knowles & Corne [8].

This algorithm was validated using several test functions taken from the specialized literature. Results were compared to the microGA for multiobjective optimization [3]. Besides graphical comparisons, two metrics were used (two set coverage and spacing). See [2] for details. Figure 1 shows a sample result for one of the problems used. In general, our algorithm had a good convergence rate to the true Pareto front, but not a very good spread of solutions. This led us to review our algorithmic design.

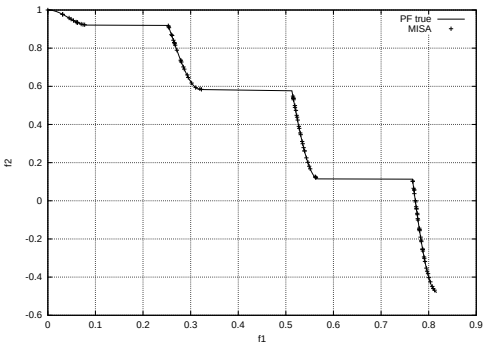

Figure 1: Comparison of results for one example. The true Pareto front is shown as a continuous line (note that the horizontal segments are NOT part of the Pareto front and are shown only to facilitate drawing the front) and the Pareto front found by our approach is shown as crosses.

4.1 Further Improvements

Our second iteration was an algorithm which departs more from the clonal selection principle but that it's more effective in practice. The main changes of our new algorithm are the following:

- All individuals from the population play the role of antibodies (i.e., there are not antigens).
- Antibodies selected for cloning are:
  - For the case of constrained problems: nondominated and feasible antibodies (dominance is measured only with respect to other feasible individuals). If there are not feasible antibodies, then all nondominated antibodies are selected.
  - For the case of unconstrained problems: all nondominated antibodies are selected.
- It uses information extracted from the secondary memory in order to determine the number of clones that must be created. The idea is to generate more clones that correspond to less populated regions of the Pareto front.

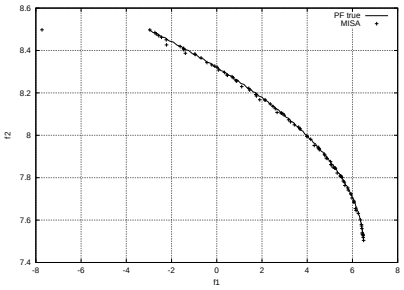

Figure 2: Pareto front obtained by our revised version of MISA in an example. The true Pareto front of the problem is shown as a continuous line.

- If the secondary memory is full, then a crossover operator is applied to its members at certain intervals, so that we can reach intermediate points between them.

Our revised algorithm (called multiobjective immune system algorithm, or MISA for short). A sample result is shown in Figure 2. In general, our revised version of MISA provided competitive results with respect to the two other algorithms against which it was compared (the NSGA-II [6] and PAES [8]). Although it did not always ranked first when using three metrics (generational distance, error ratio and spacing), in all cases it produced reasonably good approximations of the true Pareto front of each problem under study, particularly with respect to the generational distance metric, which measures closeness to the true Pareto front of the problem. See [4] for details.

5 Conclusions and Future work

Our explorations of artificial immune systems in optimization have been very fruitful so far. We have produced two highly competitive approaches: one for constrained single-objective optimization and another one for multiobjective optimization problems. Our current work focuses on further improvements to MISA. However, the main emphasis of our current work revolves around performing a theoretical study of the two algorithms described in this paper. We are particularly interested in analyzing the convergence properties of our 2 algorithms, and we are considering the use of Markov chain models for that sake.

Acknowledgements

The first author acknowledges support from CONACyT through a scholarship to pursue graduate studies at the Computer Science Section of the Electrical Engineering Department of CINVESTAV-IPN. The second author acknowledges support from CONACyT through project No. 32999-A.

References

[1] Carlos A. Coello Coello and Nareli Cruz Cortés. Use of Emulations of the Immune System to Handle Constraints in Evolutionary Algorithms. In Cihan H. Dagli et al., editor, *Intelligent Engineering Systems through Artificial Neural Networks (ANNIE'2001)*, volume 11, pages 141–146. ASME Press, July 2001.

[2] Carlos A. Coello Coello and Nareli Cruz Cortés. An Approach to Solve Multiobjective Optimization Problems Based on an Artificial Immune System. In Jonathan Timmis and Peter J. Bentley, editors, *ICARIS'2002*, pages 212–221. University of Kent at Canterbury, UK, September 2002.

[3] Carlos A. Coello Coello and Gregorio Toscano Pulido. Multiobjective Optimization using a Micro-Genetic Algorithm. In Lee Spector et al., editor, *Proceedings of GECCO'2001*, pages 274–282, San Francisco, California, 2001. Morgan Kaufmann Publishers.

[4] Nareli Cruz Cortés and Carlos A. Coello Coello. Multiobjective Optimization using the Clonal Selection Principle. In *Proceedings of GECCO'2003*, Chicago, July 2003. Springer-Verlag. (accepted).

[5] Dipankar Dasgupta, editor. *Artificial Immune Systems and Their Applications*. Springer-Verlag, Berlin, 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] P. Hajela and J. Lee. Constrained Genetic Search via Schema Adaptation. An Immune Network Solution. *Structural Optimization*, 12:11–15, 1996.

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

[9] Slawomir Koziel and Zbigniew Michalewicz. Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. *Evolutionary Computation*, 7(1):19–44, 1999.

[10] Zbigniew Michalewicz and Marc Schoenauer. Evolutionary Algorithms for Constrained Parameter Optimization Problems. *Evolutionary Computation*, 4(1):1–32, 1996.

[11] Leandro Nunes de Castro and Jonathan Timmis. *An Introduction to Artificial Immune Systems: A New Computational Intelligence Paradigm*. Springer-Verlag, 2002.

[12] Erick Cantú Paz. *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer Academic Publishers, Massachusetts, USA, 2000.