Artificial Immune System for Solving Constrained Optimization Problems

Victoria S. Aragón, Susana C. Esquivel

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

Carlos A. Coello Coello

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

Resumen

In this paper, we present an artificial immune system (AIS) based on the CLONALG algorithm for solving constrained (numerical) optimization problems. We develop a new mutation operator which produces large and small step sizes and which aims to provide better exploration capabilities. We validate our proposed approach with 13 test functions taken from the specialized literature and we compare our results with respect to Stochastic Ranking (which is an approach representative of the state-of-the-art in the area) and with respect to an AIS previously proposed by one of the co-authors.

Palabras clave:Artificial Immune System, Constrained Optimization Problems.

1 Introduction

In many real-world problems, the decision variables are subject to a set of constraints (e.g., related to the geometric properties of an object), and the search has to be bounded accordingly. Constrained optimization problems are very common, for example, in engineering applications, and therefore the importance of being able to deal with them efficiently.

Many bio-inspired algorithms (particularly evolutionary algorithms) have been very successful in the solution of a wide variety of optimization problems [14]. However, when they are used to solve constrained optimization problems, they require a suitable mechanism to incorporate constraints into their fitness functions. Within evolutionary algorithms (EAs), external penalty functions have been the most popular mechanism adopted to incorporate constraints into the fitness function [12]. However, penalty functions require the definition of accurate penalty factors (which are normally fine-tuned by hand) and the performance of the EA is highly dependent on them.

Recently, several researchers have proposed constraint-handling techniques for EAs which avoid the use of a penalty function or do not require any fine-tuning of the penalty factors [2, 7, 11]. Such approaches have been found to outperform traditional penalty functions and can handle all types of constraints (linear, nonlinear, equality, inequality).

The main motivation of the work presented in this paper is to explore the capabilities of a new mutation operator proposed on an AIS in the context of constrained global optimization. The proposed approach is based on two algorithms: (1) the CLONALG algorithm proposed by Nunes de Castro and Von Zuben [9, 10] and (2) the AIS-based approach proposed in [4].

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 and the proposed mutation operator. 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

We are interested in solving the general nonlinear programming problem which is defined as follows:

Find $\vec{x} = (x_1, \ldots, x_n)$ which optimizes

$$f(x_1, \ldots, x_n)$$

subject to:

$$h_i(x_1, \ldots, x_n) = 0 \; i = 1, \ldots, l$$

$$g_j(x_1, \ldots, x_n) \leq 0 \; j = 1, \ldots, p$$

where $(x_1, \ldots, x_n)$ is the vector of solutions (or decision variables) , l is the number of equality constraints and p is the number of inequality constraints (in both cases, constraints could be linear or nonlinear).

3 Previous Related Work

The use of artificial immune systems to solve constrained (numerical) optimization problems is scarce. The only previous related work that we found in the specialized literature is the following:

Hajela and Yoo [13, 14] have proposed a hybrid between a Genetic Algorithm (GA) and an AIS for solving constrained optimization problems. Here, the authors adopted two populations. The first is composed by the antigens (which are the best solutions), and the other by the antibodies (which are the worst solutions). The idea is to have a GA embedded into another GA. The outer GA performs the optimization of the original (constrained) problem. The second GA is run for a few generations, and uses as its fitness function a Hamming distance (binary encoding was adopted for the GA) so that the antibodies are evolved to become very similar (at the genotypic level) to the antigens, without becoming identical. One of the most interesting aspects of this work was that the infeasible individuals would normally become feasible as a consequence of the evolutionary process performed (based on similarity and not on constraint values). This approach was tested with some structural optimization problems.

Kelsey and Timmis [6] proposed an immune inspired algorithm based on the clonal selection theory to solve multimodal optimization problems. Its highlight is the mutation operator called *Somatic Contiguous Hypermutation*, where mutation is applied on a subset of contiguous bits. The length and beginning of this subset is determined randomly.

Coello Coello and Cruz-Cortés [3] have proposed an extension of Hajela and Yoo's algorithm. In this proposal, no penalty function is needed (as required by the original approach of Hajela and Yoo), and some extra mechanisms are defined to allow the approach to work in cases in which there are no feasible solutions in the initial population. Additionally, the authors proposed a parallel version of the algorithm and validated it using some standard test functions reported in the specialized literature.

Balicki [1] made a proposal very similar to the approach of Coello Coello and Cruz-Cortés. Its main difference is the way in which the antibodies' fitness is computed. In this case, Balicki introduces a ranking procedure. This approach was validated using a constrained three-objective optimization problem.

Luh and Chueh [5, 8] have proposed an algorithm (called CMOIA, or Constrained Multi Objective Immune Algorithm) for solving constrained multi-objective optimization problems. In this case, the antibody's population is composed by the potential solutions to the problem, whereas antigens are the objective functions. CMOIA transforms the constrained problem into an unconstrained one by associating an interleukine (IL) value with all the constraints violated. IL is a function of both the number of constraints violated and the total magnitude of this constraint violation (note that this IL function is actually a penalty function). Then, feasible individuals are rewarded and infeasible individuals are penalized. Other features of the approach were based on the clonal selection theory and other immunological mechanisms. CMOIA was evaluated using six test functions and two structural optimization problems.

Coello Coello and Cruz-Cortés [4] have proposed an algorithm based on the clonal selection theory for solving constrained optimization problems. The authors experimented with both binary and real-value representation, considering Gaussian-distributed and Cauchy-distributed mutations. Furthermore, they proposed a controlled and uniform mutation operator. This approach was tested with a set of 13 test functions taken from the specialized literature on evolutionary constrained optimization.

4 Our Proposed Approach

This paper presents a bio-inspired approach based on the CLONALG algorithm proposed by Nunes de Castro and Von Zuben [9, 10]. In its origins, CLONALG was used to solve pattern recognition and multimodal optimization problems, and there are few extensions of this algorithm for constrained optimization (remarkably, the approach reported in [4]).

Our proposed approach (called AIS_{const}) is another extension of CLONALG for constrained optimization, and it is described next:

1. Randomly generate j antibodies.
2. Repeat a predetermined number of times
   - 2.1. Determine the affinity of each antibody (Ab).
   - 2.2. Sort antibodies.
   - 2.3. Clone all antibodies. The antibodies are cloned proportionally to their affinities
   - 2.4. Mutate all clones.
   - 2.5. Determine the affinity of each clone.
   - 2.6. Sort clones.
   - 2.7. Select the best n individuals from the antibodies'population and the clones population.
   - 2.8. Replace the lowest affinity antibodies by new individuals generated at random.
3. End repeat.

The most relevant aspects of the approach are the following:

- All antibodies and clones are represented by vectors of real values.
- Determine the affinity of each individual (antibody or clone) implies to compute the following:
  - Feasible: an antibody is feasible if it satisfies all the constraints of the problem. All equality constraints are converted into inequality constraints, $|h(\vec{x})| - \delta \leq 0$, using a tolerance $\delta = 0.0001$, this tolerance was used by [11] and it is the value commonly used in constrained optimization.
  - Objective Function Value: objective function value for the antibody or clone.
  - Degree of constraint violation: if an antibody or clone is feasible, then its degree of constraint violation is zero. Now, if it is infeasible then its degree of constraint violation is a positive value determined by the add of $g_i(x)^+$ for $i = 1, \ldots, p$ and $|h_k(x)|$ for $k = 1, \ldots, l$.
- Antibodies are sorted using the following criterion: the feasible antibodies whose objective function are the best are placed first. Then, we place the infeasible antibodies

with the lowest degree of constraint violation. Clones are sorted using the same criterion.

- In order to select the antibodies and clones that will take part of the next iteration, we consider first the feasible individuals (over the infeasible ones) and then, those infeasible individuals that have the lowest degree of constraint violation. Note however that the best infeasible individual (from the antibodies' or clones population) will always pass to the next generation, unless the entire population is feasible. The best infeasible individual is an infeasible individual with the lowest degree of constraint violation.

- The number of clones generated from the selected antibodies is given by:

$$NC = \sum_{i=1}^{n} (int) \left(\frac{\beta * j}{i} \right)$$

  where NC is the number of clones, j is the number of antibodies and β is a multiplier factor (generally equal to 1). We used $\beta = 1$ in our experiments.

- Several mutation operators were tested, however the simplest one was ehich had the best performance and it is described next:

  If a clone is feasible, then only a single position of the string is changed for a randomly chosen value (from the allowable range for that specific decision variable).

  If a clone is infeasible, then each decision variable x_i is mutated using equation (1) or (2) (with a 50% probability).

$$x_i^{'} = x_i \pm rand(0,1) * \frac{range(x_i)}{generation} * NC \quad (1)$$

$$x_i^{'} = x_i \pm rand(0,1) * \frac{range(x_i)}{generation * NC} \quad (2)$$

  where $rand(0,1)$ refers to a random number with a uniform distribution between 0 and 1, $range(x_i)$ is a random number in the allowable range of x_i with a uniform distribution, $generation$ is the current generation number and NC is the number of clones. Equation (1) generates step sizes larger than equation (2).

4.1 Differences between AIS_{const} and the AIS proposed in [4]

There are several differences between our AIS_{const} and the AIS proposed in [4]. First, AIS_{const} makes one distinction, during the application of the mutation operator, between feasible and infeasible solutions, while the AIS proposed in [4] does not. The mutation operators of both approaches try to reduce the step sizes as the search progresses. The AIS proposed in [4] takes into account the difference between the lower and upper bounds of each decision variable, the size of the antibodies' population and their affinity. In contrast, our AIS_{const} considers the range of each decision variable, the current generation number (as the generation number grows, the mutation operator tries to reduce the step size) and the number of clones, but it only tries to reduce the step size on infeasible solutions. The main idea is that, as the search progresses, since the selection criterion is to choose individuals with the lowest degree of constraint violation for the next population, infeasible individuals that belong to the next population could be close to the boundary between the feasible and the infeasible regions.

5 Experimental Setup

In order to validate our proposed approach we tested it with a benchmark of 13 test functions taken from the specialized literature [11]. The 13 test functions are described in the Appendix at the end of this paper. The functions g02, g03, g08 and g12 are maximization problems (for simplicity, these problems were converted into minimization problems using $-f(x)$) and the rest are minimization problems.

Our results are compared with respect to Stochastic Ranking [11], which is a constraint-handling technique representative of the state-of-the-art in the area, and with respect to the AIS approach reported in [4]. 30 independents runs were performed for each problem, each consisting of 350,000 fitness function evaluations. We adopted a 20% replacement for the antibodies' population. All the statistical measures reported are taken only with respect to the runs in which a feasible solution was reached at the end.

6 Discussion of Results

Tables 1, 2 and 3 show the results obtained with the AIS proposed in [4], Stochastic Ranking and our AIS_{const}, respectively. Figures 1 to 9 show the best and mean found values for some of the functions. The description of each figure is: each column shows the best or mean value for an algorithm. The first column shows the best value found for our proposed. The second one shows the best value found for Stochastic Ranking, next it is the best value found for the AIS proposed in [4] (AIS_{former}). The three columns next show the mean found value for our AIS_{const} , Stochastic Ranking and AIS_{former}, respectively. The last column shows the optimum value.

From Table 3, we can see that our AIS_{const} was able to reach the global optimum in 3 test functions (g03, g08 and g12). Additionally, our AIS_{const} reached feasible solutions close to the global optimum in 4 more test functions (g01, g06, g09 and g11) and it found acceptable (i.e., not too far from the global optimum) feasible solutions for the rest of the test functions.

Comparing AIS_{const} with respect to Stochastic Ranking (see Tables 2 and 3), our AIS_{const} only improved the worst and mean solutions for g06. Additionally, both approaches found similar solutions for g03, g08 and g11. In the rest of the problems, Stochastic Ranking outperformed our approach.

Comparing AIS_{const} with the AIS proposed in [4] (see Tables 1 and 3), our AIS_{const} obtained better results in 3 test functions (g01, g05 and g10). However, for g05, our AIS_{const} only converged to a feasible solution in 75% of the runs while the AIS from [4] converged to a feasible solution in 90% of the runs. Both approaches found similar solutions for g03, g08 and g11. Finally, our AIS_{const} was outperformed in the remaining functions, with a difference (with respect to the best found solutions) that ranged from 0.0001 to 0.42 units. With respect to the mean and worst found solutions, our AIS_{const} was outperformed in most test functions, except for g01, g02, g08 and g12. For the last two functions both approaches found the global optimum in all runs.

Taking into account the GenMean (mean generation where the best solution was found), the fact that in none of the test functions our proposed approach got stuck in a local optimum, the small number of antibodies adopted (only 5 individuals), and the limitations imposed on the number of objective function evaluations, we argue that the mutation operator adopted by our approach is capable of performing an efficient local search over each feasible clone, which allows the algorithm to improve on the feasible solutions found. In cases in which no feasible solutions are found in the initial population, the mutation applied is capable of reaching the feasible region even when dealing with very small feasible search spaces (e.g., in g05 and g13).

Although there is clearly room for improving our proposed AIS_{const}, we have empirically shown that this approach is able of dealing with a variety of constrained optimization problems (i.e., with both linear and nonlinear constraints and objective function, and with both equality and inequality constraints). The benchmark adopted includes test functions with both small and large feasible regions, as well as a disjoint feasible region. We also argue that our proposed approach is very simple to implement and it does not require the fine-tuning of too many parameters, but only the number of antibodies to use and the percentage of replacement.

Function	Optimum	*Best*	*Mean*	*Worst*	*Std.Dev*
g01	-15	-14.9874	-14.7264	-12.9171	0.6070
g02	-0.803619	-0.8017	-0.7434	-0.6268	0.0414
g03	-1.0	-1.0	-1.0	-1.0	0.0000
g04	-30665.539	-30665.5387	-30665.5386	-30665.5386	0.0000
g05*	5126.498	5126.9990	5436.1278	6111.1714	300.8854
g06	-6961.814	-6961.8105	-6961.8065	-6961.7981	0.0027
g07	24.306	24.5059	25.4167	26.4223	0.4637
g08	-0.095825	-0.095825	-0.095825	-0.095825	0.0000
g09	680.63	680.6309	680.6521	680.6965	0.0176
g10	7049.33	7127.9502	8453.7902	12155.1358	1231.3762
g11	0.75	0.75	0.75	0.75	0.0000
g12	-1.0	-1.0	-1.0	-1.0	0.0000
g13	0.05395	0.05466	0.45782	1.49449	0.3790

Table 1: Results obtained with AIS proposed in [4]. The asterisk (*) indicates a case in which only 90% of the runs converged to a feasible solution

Function	Optimum	*Best*	*Mean*	*Worst*	*Std.Dev*	*GenMean*
g01	-15	-15.0	-15.0	-15.0	0.0E+00	741
g02	-0.803619	-0.803515	-0.781975	-0.726288	2.0E-02	1086
g03	-1.0	-1.0	-1.0	-1.0	1.9E-04	1146
g04	-30665.539	-30665.539	-30665.539	-30665.539	2.0E-05	441
g05	5126.498	5126.497	5128.881	5142.472	3.5E+00	258
g06	-6961.814	-6961.814	-6875.74	-6350.262	1.6E+02	590
g07	24.306	24.307	24.374	24.642	6.6E-02	715
g08	-0.095825	-0.095825	-0.095825	-0.095825	2.6E-17	381
g09	680.63	680.63	680.656	680.763	3.4E-02	557
g10	7049.33	7054.613	7559.192	8835.655	5.3E+02	642
g11	0.75	0.75	0.75	0.75	8.0E-05	57
g12	-1.0	-1.0	-1.0	-1.0	0.0E+00	82
g13	0.05395	0.053957	0.067543	0.216915	3.1E-02	349

Table 2: Results obtained with Stochastic Ranking [11]

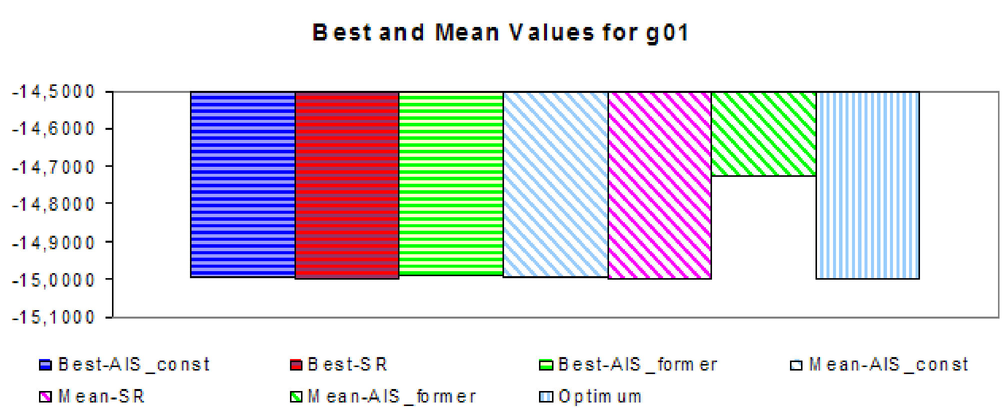

Figure 1. Best and Mean Values for g01

Function	Optimum	*Best*	*Worst*	*Mean*	*Std.Dev*	*GenMean*	*AE*
g01	-15	-14.993	-14.982	-14.989	0.002982	34180	310414
g02	-0.803619	-0.7821	-0.7230	-0.7573	0.014765	32703	340224
g03	-1.0	-1.0	-0.9108	-0.9880	0.025057	35556	347944
g04	-30665.539	-30665.1117	-30533.7827	-30645.9122	31.929167	33009	316804
g05*	5126.498	5126.660	6112.072	5468.743	339.183	28098	334444
g06	-6961.814	-6961.7940	-6956.7421	-6960.3768	1.183813	26047	135194
g07	24.306	24.531708	27.056296	25.644893	0.667470	35051	345044
g08	-0.095825	-0.095825	-0.095825	-0.095825	0.0	27071	331094
g09	680.63	680.6519	681.1474	680.8343	0.134034	34137	229054
g10	7049.33	7058.45	15787.89	8344.69	1793.850342	30739	335444
g11	0.75	0.7499	0.7499	0.7499	0.000001	25497	283304
g12	-1.0	-1.0	-1.0	-1.0	0.0	26764	345634
g13	0.05395	0.05820	16.43139	1.37142	2.904695	35095	312004

Table 3: Results obtained with our proposed AIS_{const}. The asterisk (*) indicates a case in which only 75% of the runs converged to a feasible solution. *AE* indicates the number of evaluations required to reach the best solution

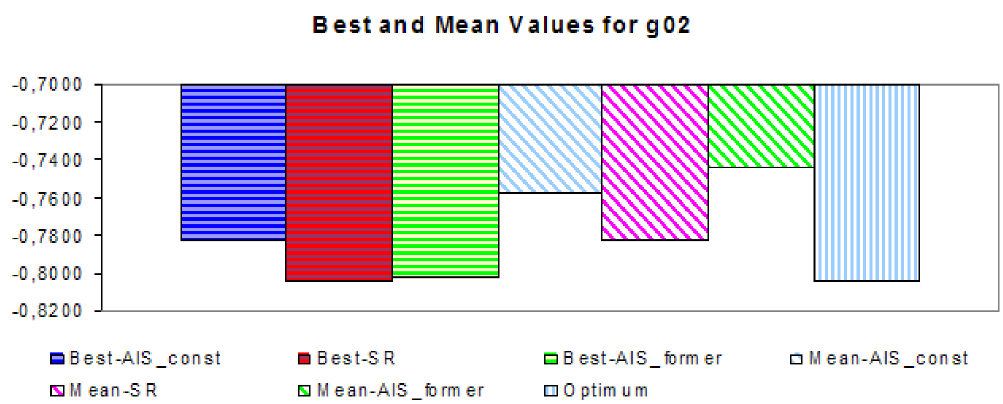

Figure 2. Best and Mean Values for g02

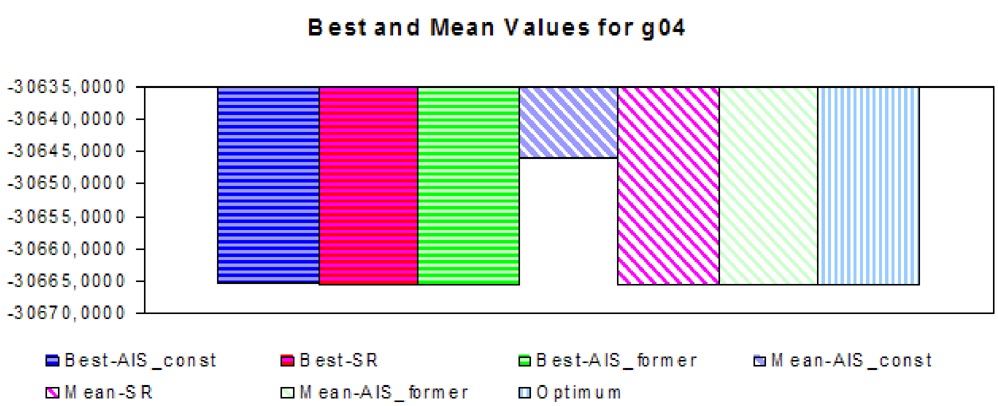

Figure 3. Best and Mean Values for g04

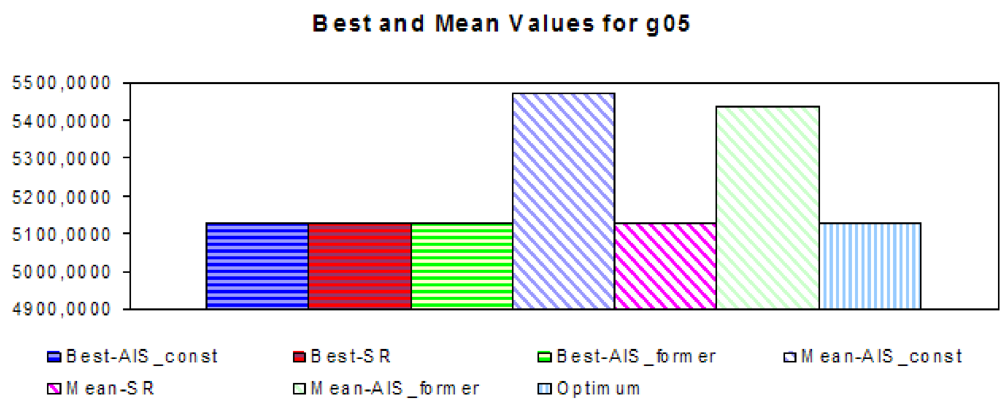

Figure 4. Best and Mean Values for g05

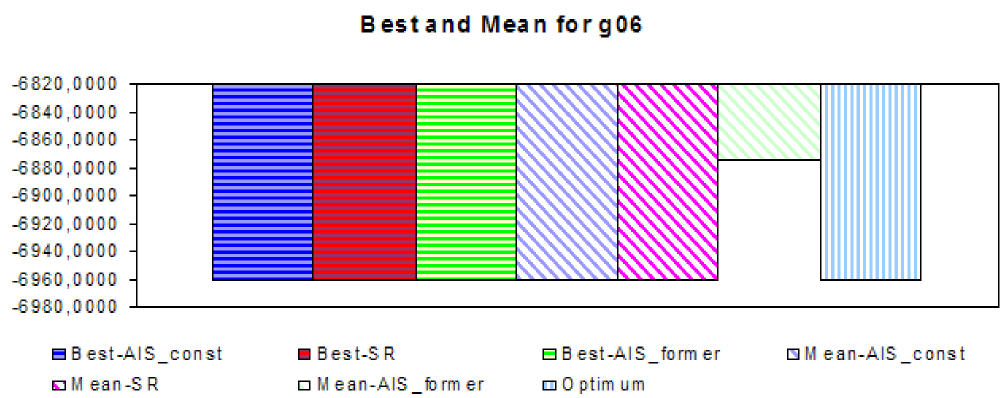

Figure 5. Best and Mean Values for g06

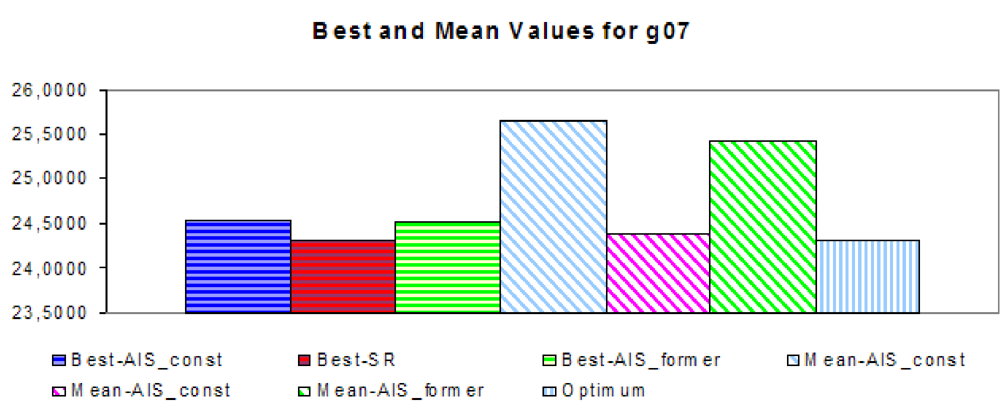

Figure 6. Best and Mean Values for g07

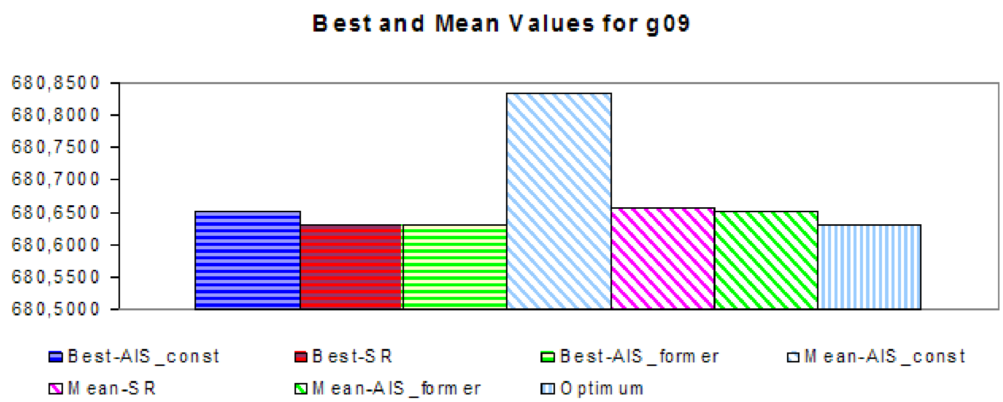

Figure 7. Best and Mean Values for g09

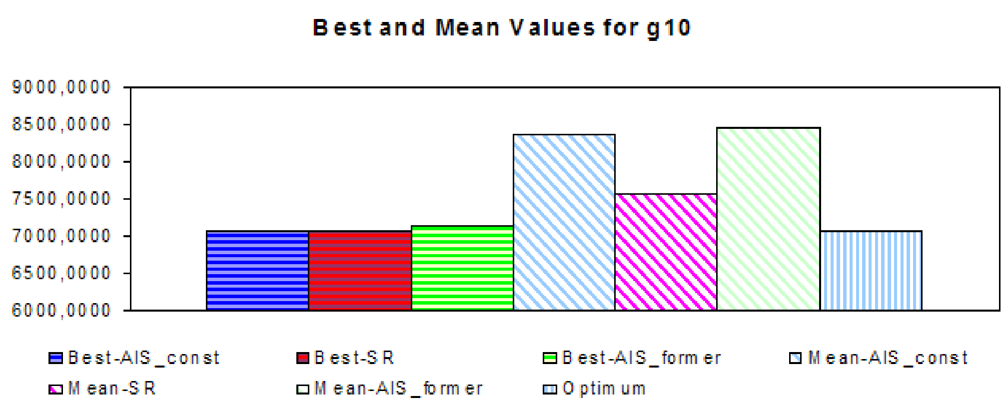

Figure 8. Best and Mean Values for g10

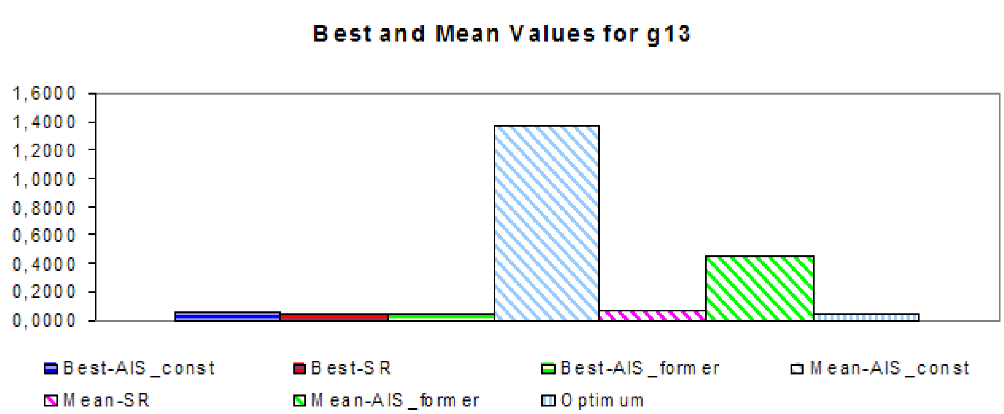

Figure 9. Best and Mean Values for g13

7 Conclusions and Future Work

This paper presents an AIS for solving constrained optimization problems in which a novel mutation operator is adopted. The approach was found to be competitive in a well-known benchmark commonly adopted in the specialized literature on constrained evolutionary optimization. The approach was also found to be robust and able to converge to feasible solutions in most cases.

Our analysis of the benchmark adopted made us realize that some test functions require small step sizes, while others require larger values. This was the motivation for proposing a mutation scheme that considers both situations.

Obviously, a lot of work remains to be done in order to improve the quality of the 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, as well as the use of boundary operators to improve the performance of our approach in problems with equality constraints. Nevertheless, it is important to emphasize that there is very little work regarding the use of artificial immune systems for constrained numerical optimization, and in that context, this approach provides a viable alternative.

Acknowledgements

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

A Test Functions

1. **g01**:
   Minimize: $f(\vec{x}) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$

   subject to:

$$
\begin{aligned}
g_1(\vec{x}) &= 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0 \\
g_2(\vec{x}) &= 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0 \\
g_3(\vec{x}) &= 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0 \\
g_4(\vec{x}) &= -8x_1 + x_{10} \leq 0 \\
g_5(\vec{x}) &= -8x_2 + x_{11} \leq 0 \\
g_6(\vec{x}) &= -8x_3 + x_{12} \leq 0 \\
g_7(\vec{x}) &= -2x_4 - x_5 + x_{10} \leq 0 \\
g_8(\vec{x}) &= -2x_6 - x_7 + x_{11} \leq 0 \\
g_9(\vec{x}) &= -2x_8 - x_9 + x_{12} \leq 0
\end{aligned}
$$

   where the bounds are $0 \leq x_i \leq 1$ ($i = 1, \ldots, 9$), $0 \leq x_i \leq 100$ ($i = 10, 11, 12$) and $0 \leq x_{13} \leq 1$. The global optimum is at $x^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)$ where $f(x^*) = -15$. Constraints g_1, g_2, g_3, g_4, g_5 and g_6 are active.

2. **g02**:
   Maximize: $f(\vec{x}) = \left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right|$ subject to:

$$
\begin{aligned}
g_1(\vec{x}) &= 0.75 - \prod_{i=1}^{n} x_i \leq 0 \\
g_2(\vec{x}) &= \sum_{i=1}^{n} x_i - 7.5n \leq 0 \qquad (3)
\end{aligned}
$$

   where $n = 20$ and $0 \leq x_i \leq 10$ ($i = 1, \ldots, n$). The global maximum is unknown; the best reported solution is [11] $f(x^*) = 0.803619$. Constraint g_1 is close to being active ($g_1 = -10^{-8}$).

3. **g03**:
   Maximize: $f(\vec{x}) = \left(\sqrt{n}\right)^n \prod_{i=1}^{n} x_i$

   subject to:

   $h(\vec{x}) = \sum_{i=1}^{n} x_i^2 - 1 = 0$

   where $n = 10$ and $0 \leq x_i \leq 1$ ($i = 1, \ldots, n$). The global maximum is at $x_i^* = 1/\sqrt{n}$ ($i = 1, \ldots, n$) where $f(x^*) = 1$.

4. **g04**:
   Minimize: $f(\vec{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$
   subject to:
   $g_1(\vec{x}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0$
   $g_2(\vec{x}) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0$
   $g_3(\vec{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0$
   $g_4(\vec{x}) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0$
   $g_5(\vec{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0$
   $g_6(\vec{x}) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0$

   where: $78 \leq x_1 \leq 102$, $33 \leq x_2 \leq 45$, $27 \leq x_i \leq 45$ ($i = 3, 4, 5$). The optimum solution is $x^* = (78, 33, 29.995256025682, 45, 36.775812905788)$ where $f(x^*) = -30665.539$. Constraints g_1 and g_6 are active.

5. **g05**

   Minimize: $f(\vec{x}) = 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3$

   subject to:
   $g_1(\vec{x}) = -x_4 + x_3 - 0.55 \leq 0$
   $g_2(\vec{x}) = -x_3 + x_4 - 0.55 \leq 0$
   $h_3(\vec{x}) = 1000\sin(-x_3 - 0.25) +$
   $1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0$

$h_4(\vec{x}) = 1000\sin(x_3 - 0.25) +$
$1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0$
$h_5(\vec{x}) = 1000\sin(x_4 - 0.25) +$
$1000\sin(x_4 - x_3 - 0.25) + 1294.8 = 0$

where $0 \le x_1 \le 1200$, $0 \le x_2 \le 1200$, $-0.55 \le x_3 \le 0.55$, and $-0.55 \le x_4 \le 0.55$. The best known solution is $x^* = (679.9453, 1026.067, 0.1188764, -0.3962336)$ where $f(x^*) = 5126.4981$.

6. **g06**
Minimize: $f(\vec{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$
subject to:
$g_1(\vec{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0$
$g_2(\vec{x}) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0$

where $13 \le x_1 \le 100$ and $0 \le x_2 \le 100$. The optimum solution is $x^* = (14.095, 0.84296)$ where $f(x^*) = -6961.81388$. Both constraints are active.

7. **g07**
Minimize: $f(\vec{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$
subject to:
$g_1(\vec{x}) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0$
$g_2(\vec{x}) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0$
$g_3(\vec{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0$
$g_4(\vec{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \le 0$
$g_5(\vec{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \le 0$
$g_6(\vec{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \le 0$
$g_7(\vec{x}) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \le 0$
$g_8(\vec{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \le 0$

where $-10 \le x_i \le 10$ $(i = 1, \ldots, 10)$. The global optimum is $x^* = (2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$ where $f(x^*) = 24.3062091$. Constraints g_1, g_2, g_3, g_4, g_5 and g_6 are active.

8. **g08**
Maximize: $f(\vec{x}) = \frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1+x_2)}$
subject to:
$g_1(\vec{x}) = x_1^2 - x_2 + 1 \le 0$
$g_2(\vec{x}) = 1 - x_1 + (x_2 - 4)^2 \le 0$

where $0 \le x_1 \le 10$ and $0 \le x_2 \le 10$. The optimum solution is located at $x^* = (1.2279713, 4.2453733)$ where $f(x^*) = 0.095825$.

9. **g09**
Minimize: $f(\vec{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$
subject to:

$g_1(\vec{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \le 0$
$g_2(\vec{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \le 0$
$g_3(\vec{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \le 0$
$g_4(\vec{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0$

where $-10 \le x_i \le 10$ $(i = 1, \ldots, 7)$. The global optimum is $x^* = (2.330499, 1.951372, -0.4775414, 4.365726, -0.6244870, 1.038131, 1.594227)$ where $f(x^*) = 680.6300573$. Two constraints are active (g_1 and g_4).

10. **g10**
Minimize: $f(\vec{x}) = x_1 + x_2 + x_3$
subject to: $g_1(\vec{x}) = -1 + 0.0025(x_4 + x_6) \le 0$
$g_2(\vec{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \le 0$
$g_3(\vec{x}) = -1 + 0.01(x_8 - x_5) \le 0$
$g_4(\vec{x}) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \le 0$
$g_5(\vec{x}) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \le 0$
$g_6(\vec{x}) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \le 0$

where $100 \le x_1 \le 10000$, $1000 \le x_i \le 10000$, $(i = 2, 3)$, $10 \le x_i \le 1000$, $(i = 4, \ldots, 8)$. The global optimum is: $x^* = (579.19, 1360.13, 5109.92, 182.0174, 295.5985, 217.9799, 286.40, 395.5979)$, where $f(x^*) = 7049.248$. g_1, g_2 and g_3 are active.

11. **g11**
Minimize: $f(\vec{x}) = x_1^2 + (x_2 - 1)^2$
subject to:
$h(\vec{x}) = x_2 - x_1^2 = 0$

where: $-1 \le x_1 \le 1$, $-1 \le x_2 \le 1$. The optimum solution is $x^* = (\pm 1/\sqrt{2}, 1/2)$ where $f(x^*) = 0.75$.

12. **g12**
Maximize: $f(\vec{x}) = \frac{100-(x_1-5)^2-(x_2-5)^2-(x_3-5)^2}{100}$
subject to:
$g_1(\vec{x}) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \le 0$

where $0 \le x_i \le 10$ $(i = 1, 2, 3)$ and $p, q, r = 1, 2, \ldots, 9$. The feasible region of the search space consists of 9^3 disjointed spheres. A point (x_1, x_2, x_3) is feasible if and only if there exist p, q, r such the above inequality (12) holds. The global optimum is located at $x^* = (5, 5, 5)$ where $f(x^*) = 1$.

13. **g13**
Minimize: $f(\vec{x}) = e^{x_1x_2x_3x_4x_5}$
subject to:

$h_1(\vec{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$
$h_2(\vec{x}) = x_2x_3 - 5x_4x_5 = 0$
$h_3(\vec{x}) = x_1^3 + x_2^3 + 1 = 0$

where $-2.3 \le x_i \le 2.3$ $(i = 1, 2)$ and $-3.2 \le x_i \le 3.2$ $(i = 3, 4, 5)$. The optimum solution is $x^* = (-1.717143, 1.595709, 1.827247, -0.7636413, -0.763645)$ where $f(x^*) = 0.0539498$.

References

[1] Jerzy Balicki. Multi-criterion Evolutionary Algorithm with Model of the Immune System to Handle Constraints for Task Assignments. In Leszek Rutkowski, Jörg H. Siekmann, Ryszard Tadeusiewicz, and Lotfi A. Zadeh, editors, *Artificial Intelligence and Soft Computing - ICAISC 2004, 7th International Conference. Proceedings*, pages 394–399, Zakopane, Poland, June 2004. Springer. Lecture Notes in Computer Science. Volume 3070.

[2] Carlos A. Coello Coello. Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, January 2002.

[3] Carlos A. Coello Coello and Nareli Cruz-Cortés. Hybridizing a genetic algorithm with an artificial immune system for global optimization. *Engineering Optimization*, 36(5):607–634, October 2004.

[4] Nareli Cruz Cortés, Daniel Trejo-Pérez, and Carlos A. Coello Coello. Handling constrained in global optimization using artificial immune system. In Christian

Jacob, Marcin L. Pilat, Peter J. Bentley, and Jonathan Timmis, editors, *Artificial Immune Systems. 4th International Conference, ICARIS 2005*, pages 234–247. Springer. Lecture Notes in Computer Science Vol. 3627, Banff, Canada, August 2005.

[5] H. Chueh G. C. Luh and W. W. Liu. MOIA: Multi Objective Immune Algorithm. *Engineering Optimization*, 35(2):143–164, 2003.

[6] Johnny Kelsey and Jon Timmis. Immune inspired somatic contiguous hypermutation for function optimisation. In Erick Cantú-Paz, James A. Foster, Kalyanmoy Deb, Lawrence Davis, Rajkumar Roy, Una-May O'Reilly, Hans-Georg Beyer, Russell K. Standish, Graham Kendall, Stewart W. Wilson, Mark Harman, Joachim Wegener, Dipankar Dasgupta, Mitchell A. Potter, Alan C. Schultz, Kathryn A. Dowsland, Natasa Jonoska, and Julian F. Miller, editors, *GECCO*, volume 2723 of *Lecture Notes in Computer Science*, pages 207–218. Springer, 2003.

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

[8] G. C. Luh and H. Chueh. Multi-objective optimal designof truss structure with immune algorithm. *Computers and Structures*, 82:829–844, 2004.

[9] L. Nunes de Castro and J. Timmis. An artificial immune network for multimodal function optimization. In *Proccedings of the 2002 Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 669–674, Honolulu, Hawaii, May 2002.

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

[11] Thomas P. Runarsson and Xin Yao. Stochastic Ranking for Constrained Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, September 2000.

[12] Alice E. Smith and David W. Coit. Constraint Handling Techniques—Penalty Functions. In Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors, *Handbook of Evolutionary Computation*, chapter C 5.2. Oxford University Press and Institute of Physics Publishing, 1997.

[13] J. Yoo and P. Hajela. Enhanced GA Based Search Through Immune System Modeling. In *3rd World Congress on Structural and Multidisciplinary Optimization*, Niagara Falls, New York, May 1999.

[14] J. Yoo and P. Hajela. Immune network modelling in design optimization. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 167–183. McGraw-Hill, London, 1999.