Constraint Handling in Genetic Algorithms via Artificial Immune Systems

Heder S. Bernardino
Universidade Federal de Juiz de Fora
Cidade Universitária
36036 330 Juiz de Fora, MG, Brazil
hedersb@gmail.com

Helio J.C. Barbosa*
Laboratório Nacional de Computação Científica
Av. Getúlio Vargas 333
25651 075 Petrópolis, RJ, Brazil
hcbm@lncc.br

Afonso C.C. Lemonge
Universidade Federal de Juiz de Fora
Departamento de Estruturas
Cidade Universitária
36036 330 Juiz de Fora, MG, Brazil
lemonge@numec.ufjf.br

ABSTRACT

The combination of an artificial immune system (AIS) with a genetic algorithm (GA) is proposed as an alternative to tackle constrained optimization problems. The AIS is inspired in the clonal selection principle and is embedded into a standard GA search engine in order to help move the population into the feasible region. The procedure is applied to well known test-problems from the evolutionary computation literature and compared to other alternative techniques.

Categories and Subject Descriptors

I.2.8 [**Problem Solving, Control Methods, and Search**]: [Heuristic methods]; J.2 [**Physical Sciences and Engineering**]: Engineering

General Terms

Algorithms

Keywords

constrained optimization, genetic algorithm, artificial immune systems

1. INTRODUCTION

Evolutionary algorithms (EAs) can be readily applied to unconstrained optimization problems by adopting a fitness function closely related to the desired objective function. However, when the solution must satisfy a set of constraints, the EA must be equiped with an additional constraint handling procedure. To fix ideas and without loss of generality, only minimization problems will be considered here.

The techniques for handling constraints within EAs can be *direct* (feasible or interior), when only feasible elements are considered, or *indirect* (exterior), when both feasible and infeasible elements are used during the search process.

*Corresponding author

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.
ACM 1-59593-186-4/06/0007.

Direct techniques comprise: a) special *closed* genetic operators[29], b) special decoders[21], c) repair techniques[25, 27], and d) "death penalty".

Direct techniques are problem dependent (with the exception of the "death penalty") and actually of extremely reduced practical applicability.

Indirect techniques include: a) the use of Lagrange multipliers[1, 2], b) the use of fitness as well as constraint violation values in a multi-objective optimization setting[31, 7], c) the use of special selection techniques[28], d) "lethalization": any infeasible offspring is just assigned a given, very low, fitness value[32], and e) penalty techniques.

Due to its simple intuitive basis and generality, penalty techniques, in spite of their shortcomings, are the most popular ones. The fitness function value of an unfeasible solution is increased by a penalty term which usually grows with the number of violated constraints and also with the amount of violation. One can have additive as well as multiplicative penalty functions. Usually, the performance of the technique depends strongly on one or more penalty parameters that must be set by the user for a given problem.

Two-parameter penalty (Le Riche et al.[20]), multi-parameter penalty (Homaifar et al.[23]), dynamically varying parameter penalty (Joines & Houck[18]), and adaptive penalty techniques (Bean & Hadj-Alouane[5], Coit et al.[30], Barbosa & Lemonge[3, 4]) are available in the literature.

Another technique which will be used here for numerical comparisons is that due to Runarsson & Yao[28] where a good balance between the objective and the penalty function values is sought by means of a stochastic ranking.

For other constraint handling methods in evolutionary computation see [29, 26, 17, 22, 19, 12, 16, 33], references therein, and the still growing literature.

However, of particular interest here is the application of ideas from artificial immune systems[10] in constrained optimization problems.

2. CONSTRAINED OPTIMIZATION PROBLEMS

A standard constrained optimization problem in R^n can be thought of as the minimization of a given objective function $f(x)$, where $x \in R^n$ is the vector of design/decision variables, subject to inequality constraints $g_p(x) \geq 0$, $p =$

$1, 2, \ldots, \bar{p}$ as well as equality constraints $h_q(x) = 0$, $q = 1, 2, \ldots, \bar{q}$. Additionally, the variables are usually subject to bounds $x_i^L \leq x_i \leq x_i^U$ which are trivially enforced in a GA and need not be considered here. Very often the design variables are further constrained to belong to a given finite set of pre-defined discrete values. This happens, for instance, in design optimization problems when parts must be selected from commercially available types with given properties. As a result, a mixed discrete-continuous constrained optimization problem arises. If further difficulties are present in the objective function, such as lack of smoothness, a GA becomes atractive. For such optimization problems arising from multidisciplinary design tasks, the constraints are in fact a complex *implicit* function of the design variables, and the check for feasibility requires an expensive computational simulation. Constraint handling techniques which do not need the explicit form of the constraints and do not require additional objective function evaluations are thus most valuable.

3. PREVIOUS WORK USING AIS

Very few papers can be found where AIS are used to solve constrained optimization problems. Those of particular interest here will be briefly considered in the following.

About ten years ago Hajela and co-workers[13, 14, 15, 34, 35] proposed the idea of using another GA embedded into the original one aiming at increasing the similarity (or reducing the distance) between infeasible elements (playing the role of antibodies) and feasible ones (antigens). The inner GA uses as fitness function a genotypical (Hamming) distance in order to evolve better (hopefully feasible) antibodies. In this way there is no need for additional expensive evaluations of the original fitness function of the problem which only happen during the search performed by the external GA. The internal GA uses a relatively inexpensive fitness based on Hamming distance calculations.

More recently, Coello and Cruz-Cortés[6] proposed an extension of Hajela's algorithm, together with a parallel version, and tested them in a larger problem set.

A different approach was followed by Cruz-Cortés et al.[8] where an existing AIS (CLONALG) (see [9, 11]) already used for pattern recognition problems and multimodal optimization is modified in order to deal with constrained optimization problems. Binary as well as real representations were considered. The results for the real coded version of CLONALG were disappointing, leading the authors to modify the mutation operator originally used, and also to remove the self-adaptation mechanism suggested in [11].

The procedure proposed here follows the idea of Hajela and co-workers in that an AIS is called to help the GA in increasing the number of feasible individuals in the population. However, instead of embedding another GA into the main search cycle, a simple technique, inspired in the clonal selection principle, is used inside the GA cycle.

4. THE PROPOSED TECHNIQUE

The proposed hybrid AIS-GA for constrained optimization consists in an outer (GA) search loop where the current population is checked for constraint violation and then divided into feasible (antigens) and infeasible individuals (antibodies). If there are no feasible individuals, the best infeasible one (that with the lowest constraint violation) is moved to the antigen population. In the following, the AIS is introduced as an inner loop where antibodies are first cloned and then mutated. Next, the distances (affinities) between antibodies and antigens are computed. Those with higher affinity (smaller sum of distances) are selected thus defining the new antibodies (closer to the feasible region). This (AIS) cycle is repeated a number of times. The resulting antibody population is then passed to the GA where constraint violations are computed as well as fitness function values for the feasible individuals. The selection operation is then performed in order to apply recombination and mutation operators to the selected parents producing a new population and finishing the external (GA) loop.

The selection procedure in the GA consists in binary tournaments where each individual is selected once and its opponent is randomly draw, with replacement, from the population. The rules of the tournament are: (i) any feasible individual is preferred to any infeasible one, (ii) between two feasible individuals, the one with the higher fitness value is chosen, and (iii) between two infeasible individuals, the one with the smaller constraint violation is chosen.

It should be noted that here the affinity is computed from the sum of phenotypical distances between individuals, employing a standard euclidean vector norm.

A pseudo-code for the proposed hybrid is given in Figure 1.

Figure 1: Pseudo-code for the Hybrid GA

```
Algorithm Hybrid AIS-GA
Begin
for i=0 to numberOfGenerationsGA do
    computeViolation();
    dividePopulation();
    antibodies <- infeasiblePop();
    antigens<- TopFeasible();
    for j=0 to numberOfIterationsAIS do
       cloneAntibodies();
       mutateAntibodies();
       computeDistanceAntibodiesAntigens();
       antibodies <- selectBetterAntibodies();
    end-do;
    computeViolationAntibodies();
    computeFitnessFeasiblePop();
    tournamentSelection();
    crossover();
    mutation();
end-do;
End
```

5. NUMERICAL EXPERIMENTS

In order to investigate the performance of the proposed hybrid procedure for constraint handling, a well know suite of function optimization problems from the literature are solved. The first 11 test functions are described in [26], and all them can be found in [28]. Thirty independent runs were performed with using a population of size 20 evolving for 17500 generations leading to 350000 function evaluations. Each real variable was encoded using 25 bits. The standard two-point crossover operator was applied with probability

$p_c = 0.8$. Mutation was applied bit-wise to the offspring with rate $p_m = 0.04$. Each equality constraint was converted into one inequality constraint bounding the absolute value of the degree of violation by 0.0001 (that is, $|h(x)| \leq 0.0001$).

The results obtained with the proposed hybrid AIS-GA with binary encoding are shown in Table 1.

The results obtained by the hybrid GA in Coello & Cortés [6], which also uses a binary encoding, are displayed in Table 2 for comparison.

The results obtained with binary encoding by the modified CLONALG technique developed by Cruz-Cortés et al[8] are displayed in Table 3

In order to give a better comparison of the proposed technique with respect to other available approaches from the literature, the results from the binary coded GAs due to Wright & Farmani[33], and those from the adaptive penalty technique (APM) obtained by Lemonge & Barbosa[24] are presented in Tables 4 and 5, respectively.

Finally, the best results from the literature, those obtained by the stochastic ranking procedure using real encoding due to Runarsson & Yao[28], are shown in Table 6.

In all Tables, INF means that the algorithm converged to an infeasible solution.

Table 1: Performance of the proposed hybrid AIS-GA.

g_i	Optimum	Best	Mean	Worst
g_1	-15	-14.9944	-14.9793	-14.9687
g_2	0.803619	0.772831	0.764654	0.756478
g_3	1	0.9989538	0.9978105	0.9933335
g_4	-30655.5	-30665.53	-30665.35	-30665.25
g_5	5126.4981	INF	INF	INF
g_6	-6961.8	-6961.804	-6961.804	-6961.804
g_7	24.306	25.373074	25.888315	26.896858
g_8	-0.095825	-0.095825	-0.095825	-0.095825
g_9	680.63	680.6817	680.7827	681.0401
g_{10}	7049.33	7320.2637	7571.3228	8081.6685
g_{11}	0.75	0.750035	0.878316	0.9992529
g_{12}	-1	-1	-1	-1
g_{13}	0.053950	INF	INF	INF

Table 2: Results obtained by the hybrid GA due to Coello & Cortés.

g_i	Optimum	Best	Mean	Worst
g_1	-15	-14.7841	-14.5266	-13.8417
g_2	0.803619	–	–	–
g_3	1	1.0046	1.0031	0.9987
g_4	-30655.5	-30665.51	-30654.98	-30517.44
g_5	5126.4981	–	–	–
g_6	-6961.8	-6961.761	-6961.273	-6960.607
g_7	24.306	–	–	–
g_8	-0.095825	–	–	–
g_9	680.63	680.9599	681.6192	683.7651
g_{10}	7049.33	–	–	–
g_{11}	0.75	–	–	–
g_{12}	-1	–	–	–
g_{13}	0.053950	–	–	–

Observing the results from the techniques which employ AIS ideas (Tables 1, 2, and 3) it is clear that the AIS-GA hybrid proposed here has better values for the best, mean, and worst results in functions g_1, g_3, g_4, g_6, g_8, g_9, and g_{10}. For functions g_2 and g_7, the modified CLONALG procedure produced a better value for the best run but a worse result for both the average and the worst run values. For function g_{11} the relative performance concerning best and mean values was inverted. However, for function g_{10} the modified CLONALG procedure was not able to produce a feasible solution. It is interesting to note that no algorithm employing AIS ideas was able to solve for both functions g_5 and g_{13}. Overall, the best performance was delivered by the AIS-GA hybrid proposed here.

Table 3: Performance of the modified CLONALG approach by Cruz-Cortés et al (binary case). Cases marked with * indicate that the algorithm converged to a feasible solution in 75% of the runs.

g_i	Optimum	Best	Mean	Worst
g_1*	-15	-14.8686	-14.6603	-12.7895
g_2	0.803619	0.775589	0.749575	0.683894
g_3*	1.0	0.99891	0.97078	0.92849
g_4	-30665.5	-30650.01	-30460.85	-30366.99
g_5	5126.498	INF	INF	INF
g_6	-6961.814	-6921.487	-6248.931	-6182.994
g_7	24.306	24.80870	30.8661	35.4455
g_8	0.095825	0.095825	0.093398	0.09313
g_9	680.630	684.12886	704.87263	753.22103
g_{10}	7049.25	INF	INF	INF
g_{11}	0.75	0.750295	0.865079	1.567670
g_{12}	1.0	0.999996	0.907750	0.725285
g_{13}	0.053950	INF	INF	INF

An additional comparison against other available binary coded procedures in the literature can be made by observing the results in Tables 4 and 5. Although the adaptive penalty technique by Lemonge & Barbosa[24] gives better results for function g_1 and g_{11}, and is also able to solve for function g_5, where the AIS-GA hybrid failed to produce a feasible solution, for the other functions the AIS-GA hybrid proposed here produced better results.

However, the best results in the literature are still those produced by the stochastic ranking procedure shown in Table 6

Table 4: Results obtained by Wright & Farmani.

g_i	Optimum	Best	Mean	Worst
g_1	-15	-14.9996	-14.84	-12.9519
g_2	0.803619	0.79434	0.76739	0.7205
g_3	1	0.99937	0.99812	0.99027
g_4	-30655.5	-30624.1	-30547.915	-30261.6
g_5	5126.4981	5126.64487	–	–
g_6	-6961.8	-6948.85	-6484.06	-6347.8
g_7	24.306	24.672	31.52044	37.98319
g_8	-0.095825	-0.09588	-0.089135	-0.0267
g_9	680.63	681.5615	688.05	712.869
g_{10}	7049.33	7298.136	8776.7699	10572.66
g_{11}	0.75	0.75	0.8151	0.9884
g_{12}	-1	–	–	–
g_{13}	0.053950	–	–	–

Table 5: **Results obtained by the Adaptive Penalty Method due to Lemonge & Barbosa.**

g_i	Optimum	Best	Mean	Worst
g_1	−15	−15	−15	−15
g_2	0.803619	0.772464	0.703197	0.600298
g_3	1	0.999391	0.975728	0.939176
g_4	−30655.5	−30665.24	−30663.40	−30660.76
g_5	5126.4981	5126.571	5389.364	6040.595
g_6	−6961.8	−6961.796	−6961.789	−6961.779
g_7	24.306	24.8637	29.8646	42.0162
g_8	−0.095825	−0.095825	−0.092616	−0.072502
g_9	680.63	680.7590	681.4076	682.1562
g_{10}	7049.33	7086.404	8161.997	10002.9
g_{11}	0.75	0.75	0.750335	0.757974
g_{12}	−1	–	–	–
g_{13}	0.053950	–	–	–

Table 6: Results from the stochastic ranking procedure of Runarsson & Yao.

g_i	Optimum	Best	Mean	Worst
g_1	−15	−15	−15	−15
g_2	0.803619	0.803515	0.7858	0.726288
g_3	1.0	1.0	1.0	1.0
g_4	−30665.539	−30665.539	−30665.539	−30665.539
g_5	5126.498	5126.497	5128.881	5142.472
g_6	−6961.814	−6961.814	−6875.940	−6350.262
g_7	24.306	24.307	24.374	24.642
g_8	0.095825	0.095825	0.095825	0.095825
g_9	680.63	680.630	680.656	680.763
g_{10}	7049.33	7054.316	7559.192	8835.655
g_{11}	0.75	0.75	0.75	0.75
g_{12}	−1.0	−1.0	−1.0	−1.0
g_{13}	0.05395	0.053957	0.067543	0.216915

6. CONCLUSIONS

A hybrid artificial immune system aided genetic algorithm was proposed and tested in a well known set of constrained optimization problems of the evolutionary computation literature. Comparison with some alternative approaches were performed.

Overall, the best performance among AIS inspired procedures was delivered by the AIS-GA hybrid proposed here. When compared to other binary coded techniques the AIS-GA also produced better results except for functions g_1 and g_5.

However, the best results in the literature are still those produced by the stochastic ranking procedure, although one should note that part of its success is probably due to the real encoding and operators adopted which are more adequate to continuous function optimization than the binary encodings and operators used here.

It is interesting to note that no algorithm employing AIS ideas was able to solve for two functions in the set (g_5 and g_{13}), and this should be further investigated.

It is clear that AIS-aided constraint handling techniques in general need further research, which also applies to the promising AIS-GA hybrid proposed here

7. ACKNOWLEDGMENTS

The authors thank CNPq (grant no. 302299/2003-3), and MCT/LNCC/PRONEX.

8. REFERENCES

[1] H. Adeli and N.-T. Cheng. Augmented Lagrangian Genetic Algorithm for Structural Optimization. *Journal of Aerospace Engineering*, 7(1):104–118, January 1994.

[2] H. J. C. Barbosa. A Coevolutionary Genetic Algorithm for Constrained Optimization. In *Proceedings of the Congress on Evolutionary Computation 1999 (CEC'99)*, volume 3, pages 1605–1611. IEEE Service Center, July 1999.

[3] H. J. C. Barbosa and A. C. C. Lemonge. An adaptive penalty scheme in genetic algorithms for constrained optimization problems. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002)*, pages 287–294, July 2002.

[4] H. J. C. Barbosa and A. C. C. Lemonge. An Adaptive Penalty Scheme for Steady-State Genetic Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2003)*, pages 718–729. Springer Verlag, July 2003.

[5] J. Bean and A. Alouane. A dual genetic algorithm for bounded integer programs. Technical Report TR 92-53, Departament of Industrial and Operations Engineering, The University of Michigan, 1992.

[6] C. A. C. Coello and N. C. C. es. Hybridizing a genetic algorithm with an artificial immune system for global optimization. *Engineering Optimization*, 36(5):607–634, October 2004.

[7] C. A. C. Coello and E. Mezura-Montes. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*, 16(3):193–203, July 2002.

[8] N. C. Cortés, D. Trejo-Pérez, and C. A. C. Coello. Handling constraints in global optimization using an artificial immune system. In *ICARIS*, volume 3627 of *Lecture Notes in Computer Science*, pages 234–247. Springer, 2005.

[9] L. N. de Castro and J. Timmis. An artificial immune network for multimodal function optimization. In *Proc. of the 2002 Congress on Evolutionary Computation,*, volume I, pages 669–674, Honolulu, Hawaii, USA, May 2002.

[10] L. N. de Castro and J. Timmis. *An Introduction to Artificial Immune Systems: A New Computational Intelligence Paradigm*. Springer-Verlag, 2002.

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

[12] K. Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2-4):311–338, June 2000.

[13] P. Hajela and J. Lee. Constrained Genetic Search via Schema Adaptation. An Immune Network Solution. In N. Olhoff and G. I. N. Rozvany, editors, *Proceedings of the First World Congress of Stuctural and Multidisciplinary Optimization*, pages 915–920, Goslar, Germany, 1995. Pergamon.

[14] P. Hajela and J. Lee. Constrained genetic search via schema adaptation. An immune network solution. *Structural Optimization*, 12:11–15, 1996.

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

[16] S. B. Hamida and M. Schoenauer. An Adaptive Algorithm for Constrained Optimization Problems. In *Proceedings of 6th Parallel Problem Solving From Nature (PPSN VI)*, pages 529–538, Heidelberg, Germany, September 2000. Springer-Verlag. Lecture Notes in Computer Science Vol. 1917.

[17] R. Hinterding and Z. Michalewicz. Your brains and my beauty: Parent matching for constrained optimization. In *Proc. of the Fifth Int. Conf. on Evolutionary Computation*, pages 810–815, Alaska, May 4-9 1998.

[18] J. Joines and C. R. Houck. On the use of non-stationary penalty methods to solve nonlinear constrained optimization problems with GAs. In D. Fogel and Z. Michalewicz, editors, *Proc. of 1994 IEEE Conf. on Evolutionary Computation*, pages 579–585, Piscataway, New Jersey: IEEE, 1994.

[19] J.-H. Kim and H. Myung. Evolutionary programming techniques for constrained optimization problems. *IEEE Transactions on Evolutionary Computation*, 2(1):129–140, 1997.

[20] R. L. R. C. Knopf-Lenoir and R. Haftka. A segregated genetic algorithm for constrained structural optimization. In L. Eshelman, editor, *Proc. of the Sixth Int. Conf. on Genetic Algorithms*, pages 558–565, Pittsburgh, PA., July 1995.

[21] S. Koziel and Z. Michalewicz. A Decoder-based Evolutionary Algorithm for Constrained Parameter Optimization Problems. In *Proceedings of the 5th Parallel Problem Solving from Nature (PPSN V)*, pages 231–240. Springer-Verlag, September 1998. Lecture Notes in Computer Science Vol. 1498.

[22] S. Koziel and Z. Michalewicz. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. *Evolutionary Computation*, 7(1):19–44, 1999.

[23] H. H. S.-Y. Lai and X. Qi. Constrained optimization via genetic algorithms. *Simulation*, 62(4):242–254, 1994.

[24] A. Lemonge and H. Barbosa. An Adaptive Penalty Scheme for Genetic Algorithms in Structural Optimization. *International Journal for Numerical Methods in Engineering*, 59(5):703–736, February 2004.

[25] G. Liepins and W. Potter. A Genetic Algorithm Approach to Multiple-Fault Diagnosis. In L. Davis, editor, *Handbook of Genetic Algorithms, chapter 17*, pages 237–250, Van Nostrand Reinhold, New York, NY., 1991.

[26] Z. Michalewicz and M. Shoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4(1):1–32, 1996.

[27] D. Orvosh and L. Davis. Using a Genetic Algorithm to Optimize Problems with Feasibility Contraints. In *Proc. of the First IEEE Conference on Evolutionary Computation*, pages 548–553, IEEE Press, 1994.

[28] T. P. Runarsson and X. Yao. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, September 2000.

[29] M. Shoenauer and Z. Michalewicz. Evolutionary computation at the edge of feasibility. In H.-M. V. W. E. I. Rechenberg and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN IV*, volume 1141, pages 245–254, Berlin, 1996. Springer-Verlag. LNCS.

[30] D. C. A. Smith and D. Tate. Adaptive penalty methods for genetic optimization of constrained combinatorial problems. *INFORMS Journal on Computing*, 6(2):173–182, 1996.

[31] P. D. Surry and N. J. Radcliffe. The COMOGA Method: Constrained Optimisation by Multiobjective Genetic Algorithms. *Control and Cybernetics*, 26(3):391–412, 1997.

[32] A. von Kampen, C. Strom, and L. Buydens. Lethalization, penalty and repair functions for constraint handling in the genetic algorithm methodology. *Chemometrics and Intelligent Laboratory Systems*, 34:55–68, 1996.

[33] J. Wright and R. Farmani. Genetic Algorithms: A Fitness Formulation for Constrained Minimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 725–732, San Francisco, California, July 2001. Morgan Kaufmann Publishers.

[34] J. Yoo and P. Hajela. Immune network simulations in multicriterion design. *Structural Optimization*, 18:85–94, 1999.

[35] J. Yoo and P. Hajela. Immune network simulations in multicriterion design. *Structural Optimization*, 18:85–94, 1999.