A hybrid genetic algorithm for constrained optimization problems in mechanical engineering

Conference Paper · September 2007

DOI: 10.1109/CEC.2007.4424532 · Source: DBLP

3 authors:

Heder Bernardino
Federal University of Juiz de Fora

Helio Barbosa
Laboratório Nacional de Computação Científica

Afonso C. C. Lemonge
Federal University of Juiz de Fora

A Hybrid Genetic Algorithm for Constrained Optimization Problems in Mechanical Engineering

H.S. Bernardino
UFJF
Campus Universitário,
Juiz de Fora, MG, Brazil
CEP 36036-330,
Email: hedersb@gmail.com

H.J.C. Barbosa
LNCC/MCT
Av. Getúlio Vargas, 333,
Petrópolis, RJ, Brazil.
CEP 25651-075
Email: hcbm@lncc.br

A.C.C. Lemonge
UFJF
Campus Universitário,
Juiz de Fora, MG, Brazil
CEP 36036-330,
Email: afonso.lemonge@ufjf.edu.br

***Abstract*— A genetic algorithm (GA) is hybridized with an artificial immune system (AIS) as an alternative to tackle constrained optimization problems in engineering. 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 mechanical engineering problems available in the literature and compared to other alternative techniques.**

I. Introduction

Evolutionary algorithms (EAs), which can be readily applied to unconstrained optimization problems, must be equipped with an additional constraint handling procedure every time that the constraints cannot be automatically satisfied by all candidate solutions in the population.

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.

Direct techniques comprise: a) special *closed* genetic operators[1], b) special decoders[2], c) repair techniques[3], 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[4], [5], b) combining fitness and constraint violation in a multi-objective optimization setting[6], [7], c) the use of special selection techniques[8], d) assigning to any infeasible offspring a very low fitness value[9], and e) penalty techniques[10], [11], [12], [13], [14], [15], [16].

For other constraint handling methods in evolutionary computation see [1], [17], [18], [2], [19], [20], [21], [22], references therein, and the still growing literature.

However, of particular interest here is the application of ideas from artificial immune systems (AIS) in constrained optimization problems. A hybrid Genetic Algorithm is proposed to solve constrained optimization problems in mechanical engineering. An additional technique, called Clearing, is used in order to improve the quality of the results obtained by the proposed hybrid GA. This paper is organized as follows. The formulation of the constrained optimization problem is described in Section II, previous works using AIS are presented in Section III. The proposed technique is given in Section IV, numerical experiments are discussed in Section V, and, finally, Section VI presents some conclusions.

II. 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 values, as in design optimization problems when parts must be selected from commercially available types. A mixed discrete-continuous constrained optimization problem arises. 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 require the explicit form of the constraints and do not require additional objective function evaluations are thus most valuable.

III. Previous Work Using AIS

Not many 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[23], [24], [25], [26] 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.

1-4244-1340-0/07/$25.00 ©2007 IEEE

More recently, Coello and Cruz-Cortés[27] 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.[28] where an existing AIS (CLONALG) (see [29], [30]) 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 [30].

IV. The Proposed Technique

In a previous work[31], following the idea of Hajela and co-workers, a hybrid GA was proposed where 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. 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 two better infeasible ones (those with the lowest constraint violation) are moved to the antigen population. The number of copies of better infeasible individuals can be set by the user. 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 with the same fitness already calculated. 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 genotypical distances between individuals, employing the standard Hamming distance.

A pseudo-code for the proposed hybrid is given in Algorithm 1 and some auxiliary functions in Algorithms 2 and 3.

Petrowski's clearing procedure [32], originally used for multimodal problems, is a niching method inspired by the principle of sharing limited resources within subpopulations

Algorithm 1 The Hybrid GA Algorithm

```
procedure HYBRIDGA(nGenGA,nIterAIS)
    COMPUTEFITNESSVIOLATION(population)
    for i = 1 : nGenerationsGA do
        DIVIDE(population, antibodies, antigens)
        for j = 1 : nIterationsAIS do
            CLONE(antibodies, temp)
            MUTATION(temp)
            COMPUTEDISTANCE(antigens, temp)
            SELECTBETTER(temp, antibodies)
        end for
        UNION(antibodies, antigens, population)
        TOURNAMENTSELECTION(population, temp)
        CROSSOVER(temp)
        MUTATION(temp)
        COMPUTEFITNESSVIOLATION(temp)
        CHANGEPOPULATION(population, temp)
    end for
end procedure
```

Algorithm 2 Auxiliary Functions

```
function CLONE(antibodies, temp)
    temp ← antibodies
    for i = 1 : numClones do
        for j = 1 : antibodies.size do
            ADD(temp, antibodies[j])
        end for
    end for
end function

function SELECTBETTER(temp, antibodies)
    CLEAR(antibodies)
    for i = 1 : temp.size do
        ADD(tmp, temp[j])
        if i mod numClones = 0 then
            GETBEST(tmp, antibody)
            ADD(antibodies, antibody)
            CLEAR(tmp)
        end if
    end for
end function

function TOURNAMENTSELECTION(population, temp)
    CLEAR(temp)
    for i = 1 : population.size do
        RAMDOM(r)
        GETBEST(population[i], population[r], best)
        ADD(temp, best)
    end for
end function
```

Algorithm 3 changePopulation

```
function CHANGEPOPULATION^C(population, temp)
    UNION(population, temp, tmp)
    SORT(tmp)
    for i = 1 : tmp.size do
        for j = i + 1 : tmp.size do
            if not ISCLEARING(tmp[j]) then
                CALCDISTANCE(tmp[i], tmp[j], d)
                if d < criticalDistance then
                    SETCLEAR(tmp[j])
                end if
            end if
        end for
    end for
    CLEAR(population)
    for i = 1 : tmp.size do
        if not ISCLEARING(tmp[i]) then
            if temp.size! = population.size then
                ADD(population, tmp[i])
            end if
        end if
    end for
    SORT(temp)
    i ← 1
    while temp.size! = population.size do
        if ISCLEARING(tmp[i]) then
            ADD(population, temp[i])
            i ← i + 1
        end if
    end while
end function
```

of individuals characterized by some similarities [33]. The clearing procedure leaves those resources to the better individuals of each subpopulation. According to [33], that procedure is normally applied after evaluating the fitness of individuals and before applying the selection operator. The individuals are sorted from best to worst and all solutions having a critical distance from each pivot solution in the population have their fitness values set to zero. The pivot is the best individual not cleared in the sequence. This procedure is continued until all solutions are considered, that is either to be a pivot or to be cleared.

Differently from [33], the clearing procedure is applied here when a new population is substituted for the previous one. A new set of individuals is created from the union of both populations (previous and next populations). The procedure of clearing is then executed on that union. The fitness values are not set to zero as in [33]. Instead, the individuals cleared are tagged. The new population is made up of non-cleared individuals and, if necessary, completed with the best cleared individuals generated from crossover and mutation.

In [33], the clearing procedure when applied alone did not produce good results. In order to keep the niches the crossover operator is applied here to similar individuals.

The remaining steps of the technique proposed here are not changed.

V. Numerical Experiments

In order to investigate the performance of the proposed algorithm, six mechanical engineering optimization problems often discussed in the literature are considered in the following. For the AIS-GA presented in this paper the numerical experiments use a population size equal to 20, a binary Gray code with 50 bits for each continuous variable, a crossover probability equal to 1, a mutation rate of 0.02, elitism (the 2 best individuals are copied to the next generation), a maximum of 20 iterations ($nIterationsAIS = 20$) of the AIS, the number of clones set to 3 ($numClones = 3$), and, finally, the radius ($criticalDistance$) of the clearing procedure (when it is applied) was set to 10% of the length of the chromosome.

A. The Tension/Compression String Design

This example corresponds to the minimization of the volume V of a coil spring, depicted in the Figure 1, under a constant tension/compression load. There are three design variables to be considered: The number $x_1 = N$ of active coils of the spring, the winding diameter $x_2 = D$ and the wire diameter $x_3 = d$. The volume of the coil to be minimized is written as [34]:

$$V(x) = (x_1 + 2)\, x_2 x_3^2$$

and is subject to the constraints

$$\begin{aligned} g_1(x) &= 1 - \frac{x_2^3 x_1}{71785 x_3^4} \leq 0 \\ g_2(x) &= \frac{4x_2^2 - x_3 x_2}{12566(x_2 x_3^3 - x_3^4)} + \frac{1}{5108 x_3^2} \leq 0 \\ g_3(x) &= 1 - \frac{140.45 x_3}{x_2^2 x_1} \leq 0 \\ g_4(x) &= \frac{x_2 + x_3}{1.5} - 1 \leq 0 \end{aligned}$$

where

$$2 \leq x_1 \leq 15 \qquad 0.25 \leq x_2 \leq 1.3 \qquad 0.05 \leq x_3 \leq 2$$

A comparison of results is provided in the Table I where the best result is found by the AIS-GA with clearing, presenting a final volume equal to 0.012666. The Table II shows the values found for the design variables and constraints corresponding to the best solution for the Tension/Compression String design. The reference [34] did not present the final values of the design variables for this problem. The number of function evaluations was set equal to 36,000 for all experiments.

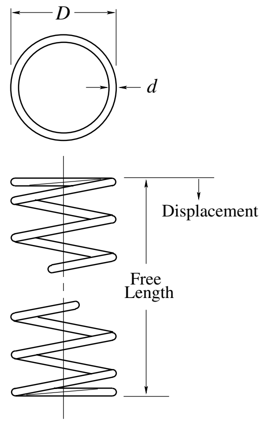

Fig. 1. The Tension/Compression String

TABLE I
VALUES FOUND FOR TENSION/COMPRESSION STRING DESIGN WHERE THE SUPERSCRIPT (C) DENOTES THE AIS-GA WITH CLEARING

	Best	Average	Worst
Ref. [34]	0.012688	0.013014	0.017037
AIS-GA	0.012668	0.013481	0.016155
AIS-GA^C	**0.012666**	**0.012974**	**0.013880**

B. The Speed Reducer design

The objective of this problem is to minimize the weight W of the speed reducer [34] shown in the Figure 2. The design variables are the face width ($x_1 = b$), the module of teeth ($x_2 = m$), the number of teeth on pinion ($x_3 = n$), the length of the shaft 1 between the bearings ($x_4 = l_1$), the length of the shaft 2 between the bearings ($x_5 = l_2$), the diameter of the shaft 1 ($x_6 = d_1$), and, finally, the diameter of the shaft 2 ($x_7 = d_2$). The third variable is integer and all the others are continuous. The constraints include limitations on the bending and surface stress of the gear teeth, transverse deflections of the shafts 1 and 2 generated by the transmitted force, and, finally, the stress in the shafts 1 and 2. The weight

TABLE II
VALUES FOUND FOR THE DESIGN VARIABLES AND CONSTRAINTS FOR THE TENSION/COMPRESSION STRING DESIGN WHERE nfe DENOTES THE TOTAL NUMBER OF FUNCTION EVALUATIONS

Var.	AIS-GA	AIS-GA^C
x_1	11.852177	11.329555
x_2	0.34747463	0.35603234
x_3	0.051301897	0.051660806
g_1	-0.00000012	-0.000006437
g_2	-0.00000047	-0.000013709
g_3	-4.03513200	-4.052324300
g_4	-0.73414900	-0.728204600
V	0.012668	**0.012666**
nfe	36,000	36,000

of the speed reducer, to be minimized, is given by

$$\begin{aligned} W(x) = \quad & 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) \\ & -1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) \\ & +0.7854\left(x_4x_6^2 + x_5x_7^2\right) \end{aligned}$$

subject to

$$\begin{aligned} & g_1(x) = 27x_1^{-1}x_2^{-2}x_3^{-1} \le 1 \\ & g_2(x) = 397.5x_1^{-1}x_2^{-2}x_3^{-2} \le 1 \\ & g_3(x) = 1.93x_2^{-1}x_3^{-1}x_4^3x_6^{-4} \le 1 \\ & g_4(x) = 1.93x_2^{-1}x_3^{-1}x_5^3x_7^{-4} \le 1 \\ & g_5(x) = \frac{1}{0.1x_6^3}\left[\left(\frac{745x_4}{x_2x_3}\right)^2 + \{16.9\}\,10^6\right]^{0.5} \le 1100 \\ & g_6(x) = \frac{1}{0.1x_7^3}\left[\left(\frac{745x_5}{x_2x_3}\right)^2 + (157.5)\,10^6\right]^{0.5} \le 850 \\ & g_7(x) = x_2x_3 \le 40 \\ & g_8(x) = x_1/x_2 \ge 5 \\ & g_9(x) = x_1/x_2 \le 12 \\ & g_{10}(x) = (1.5x_6 + 1.9)\,x_4^{-1} \le 1 \\ & g_{11}(x) = (1.1x_7 + 1.9)\,x_5^{-1} \le 1 \\ & 2.6 \le x_1 \le 3.6 \quad 0.7 \le x_2 \le 0.8 \quad 17 \le x_3 \le 28 \\ & 7.3 \le x_4 \le 8.3 \quad 7.8 \le x_5 \le 8.3 \quad 2.9 \le x_6 \le 3.9 \end{aligned}$$

The Table III presents a comparison of results found by the proposed algorithm and those given in the references [34] and [27]. The AIS-GA and AIS-GA with clearing found essentially the same values presented in the reference [27], and are better than those found in [34]. Furthermore, the AIS-GA used 36,000 functions evaluations (as in [34]), whereas the results presented in [27] were reached using 150,000 function evaluations.

Table IV presents the best final values of the design variables and constraints for the Speed Reducer design. In [27] the result for the best weight is given as 2994.3419. However, using the design variables presented in that reference, the value of the weight found is equal to 2994.4717 marked with an * in Table III. The weight found by the AIS-GA^C is equal to 2994.4712.

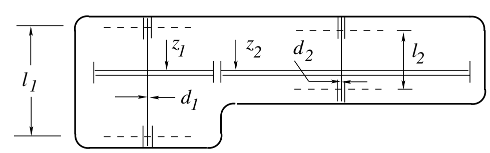

Fig. 2. The Speed Reducer

C. The Welded Beam design

This test corresponds to the design of the welded beam depicted in the Figure 3. The design variables are $\{h, l, t, b\}$, with bounds $0.125 \le h \le 10$, and $0.1 \le l, t, b \le 10$.

TABLE III

VALUES FOUND FOR THE SPEED REDUCER DESIGN

	nfe	Best	Average	Worst
Ref. [34]	$36,000$	3025.0051	3088.7778	3078.5918
AIS-GA	$36,000$	2994.4720	2994.4836	2994.5090
AIS-GA^C	$36,000$	**2994.4712**	**2994.4712**	**2994.4712**
Ref. [27]	$150,000$	**2994.3419**	**2994.3472**	**2994.3768**
AIS-GA	$150,000$	2994.4712	2994.4712	2994.4712
AIS-GA^C	$150,000$	2994.4712	2994.4712	2994.4712
Ref. [27]*	$150,000$	2994.4717	–	–

TABLE IV

VALUES FOUND FOR THE SPEED REDUCER DESIGN

Var	Ref. [34]	Ref. [27]	AIS-GA	AIS-GA^C
x_1	3.506163	3.500000	3.500001	3.5
x_2	0.700831	0.700000	0.700000	0.7
x_3	17.0	17.0	17.0	17.0
x_4	7.460181	7.300008	7.300017	7.3000035
x_5	7.962143	7.715322	7.715326	7.7153225
x_6	3.362900	3.350215	3.350216	3.3502147
x_7	5.308949	5.286655	5.286654	5.2866545
g_1	-0.077734	-0.07391524	-0.07391554	-0.07391524
g_2	-0.201305	-0.19799852	-0.19799876	-0.19799852
g_3	-0.474119	-0.49917084	-0.49916983	-0.49917156
g_4	-0.897068	-0.90464383	-0.90464365	-0.90464383
g_5	-0.011021	-2.3×10^{-7}	-1.55×10^{-6}	-1.19×10^{-7}
g_6	-0.012500	-2.9×10^{-7}	0.00000000	0.00000000
g_7	-0.702147	-0.70250000	-0.70250000	-0.70250000
g_8	-0.000573	0.00000000	-2.9×10^{-7}	0.00000000
g_9	-0.583095	-0.5833333	-0.58333320	-0.58333330
g_{10}	-0.069144	-0.051326692	-0.05132753	-0.05132616
g_{11}	-0.027920	-0.00000018	-0.00000077	-0.00000036
W	3025.0051	**2994.3419**	2994.4720	**2994.4712**
nfe	36,000	150,000	36,000	36,000

The objective function to be minimized is the cost C of the beam given as:

$$C(h,l,t,b) = 1.10471h^2l + 0.04811tb(14.0 + l)$$

subject to

$$g_1(\tau) = 13,600 - \tau \geq 0 \qquad g_2(\sigma) = 30,000 - \sigma \geq 0$$
$$g_3(b,h) = b - h \geq 0 \qquad g_4(P_c) = P_c - 6,000 \geq 0$$
$$g_5(\delta) = 0.25 - \delta \geq 0$$

The expressions for τ, σ, P_c, and δ are given by:

$$\tau = \sqrt{(\tau')^2 + (\tau'')^2 + l\tau'\tau''/\alpha}$$
$$\alpha = \sqrt{0.25(l^2 + (h+t)^2)} \qquad \sigma = \frac{504000}{t^2b}$$
$$P_c = 64746.022(1 - 0.0282346t)tb^3$$
$$\delta = \frac{2.1952}{t^3b} \qquad \tau' = \frac{6000}{\sqrt{2}hl}$$
$$\tau'' = \frac{6000(14 + 0.5l)\alpha}{2(0.707hl(l^2/12 + 0.25(h+t)^2))}$$

The Table V shows a comparison of results with the algorithms proposed here and a genetic algorithm approach using an adaptive penalty method presented in [35]. The best results found correspond to the AIS-GA with clearing. The

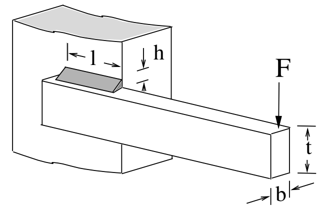

Fig. 3. The Welded Beam

Table VI shows the design variables and constraint values corresponding to the best solution found by each technique. The number of function evaluations was set equal to 320,000.

TABLE V

VALUES FOUND FOR THE COST OF THE WELDED BEAM DESIGN.

	Best	Average	Worst
Ref. [35]	2.38159	2.41718	2.95533
AIS-GA	2.38125	2.59303	3.23815
AIS-GA^C	**2.38122**	**2.38992**	**2.41391**

TABLE VI

RESULTS FOR THE DESIGN VARIABLES AND CONSTRAINTS WITH RESPECT TO THE BEST SOLUTIONS OF THE WELDED BEAM DESIGN.

Var.	Ref. [35]	AIS-GA	AIS-GA^C
h	0.2442949	0.24432427	0.24438575
l	6.2116738	6.2201996	6.2183037
t	8.3015486	8.291464	8.291165
b	0.2443003	0.24436942	0.24438748
g_1	0.0004447	0.000000000	0.001953125
g_2	64.378068	0.001953125	0.056640625
g_3	0.0000054	0.000045150	0.000001728
g_4	0.0002553	0.029785156	1.210937500
g_5	0.2342937	0.234240830	0.234240280
$Cost$	2.38159	2.381246	**2.3812175**
nfe	320,000	320,000	320,000

D. The Pressure Vessel design

This problem, often studied in the literature [36], [37], [38], [39], corresponds to the weight minimization of a cylindrical pressure vessel with two spherical heads as shown in Figure 4. The objective function involves four variables: the thickness of the pressure vessel (T_s), the thickness of the head (T_h), the inner radius of the vessel (R) and the length of the cylindrical component (L). Since there are two discrete variables (T_s and T_h) and two continuous variables (R and L), one has a nonlinearly constrained mixed discrete-continuous optimization problem. The bounds of the design variables are $0.0625 \leq T_s, T_h \leq 5$ (in constant steps of 0.0625) and $10 \leq R, L \leq 200$. The design variables are given in inches and the weight is written as:

$$W(T_s,T_h,R,L) = 0,6224T_sT_hR + 1.7781T_hR^2 + 3.1661T_s^2L + 19.84T_s^2R$$

to be minimized subject to the constraints

$$g_1(T_s, R) = T_s - 0.0193R \geq 0$$
$$g_2(T_h, R) = T_h - 0.00954R \geq 0$$
$$g_3(R, L) = \pi R^2 L + 4/3\pi R^3 - 1,296,000 \geq 0$$
$$g_4(L) = -L + 240 \geq 0$$

The first two constraints establish a lower bound to the ratios T_s/R and T_h/R, respectively. The third constraint corresponds to a lower bound for the volume of the vessel and the last one to an upper bound for the length of the cylindrical component. The Table VII makes a comparison

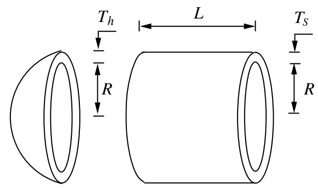

Fig. 4. The Pressure Vessel.

of results obtained with the algorithms proposed in this paper, and some results from the literature. The algorithms AIS-GA in this paper and the GA in [35] used 80,000 against 150,000 function evaluations in [27]. The best solution was found by the AIS-GA with clearing and corresponds to a final weight of 6060.138. The Table VIII shows the details of the final best solutions.

TABLE VII

VALUES OF THE WEIGHT FOUND FOR THE PRESSURE VESSEL DESIGN.

	Best	Average	Worst
Ref. [27]	6061.123	6734.085	7368.060
Ref. [35]	6060.188	**6311.766**	**6838.939**
AIS-GA	6060.368	6743.872	7546.750
AIS-GA^C	**6060.138**	6385.942	6845.496

TABLE VIII

DESIGN VARIABLES, CONSTRAINTS AND WEIGHT FOUND FOR THE PRESSURE VESSEL DESIGN

Var.	Ref. [27]	Ref. [35]	AIS-GA	AIS-GA^C
T_s	0.8125	0.8125	0.8125	0.8125
T_h	0.4375	0.4375	0.4375	0.4375
R	42.086994	42.0946558	42.093082	42.094967
L	176.779128	176.684062	176.70308	176.67972
g_1	0.000221	0.000073	0.0001035	0.000007
g_2	0.035990	0.035917	0.0359320	0.035914
g_3	3.219817	2.929000	0.1562500	0.0625
g_4	63.220872	63.315938	63.296920	63.320282
W	6061.1229	6060.187934	6060.3677	**6060.138**
nfe	150,000	80,000	80,000	80,000

E. The Cantilever Beam design

This test problem[40] corresponds to the minimization of the volume of the cantilever beam shown in the Figure 5 where the load P is equal to 50000 N. There are 10 design variables corresponding to the height (H_i) and width (B_i) of the rectangular cross-section of each of the five constant steps shown in the Figure 5. The variables B_1 and H_1 are integer, B_2 and B_3 assume discrete values to be chosen from the set $\{2.4, 2.6, 2.8, 3.1\}$, H_2 and H_3 are discrete and chosen from the set $\{45.0, 50.0, 55.0, 60.0\}$ and, finally, B_4, H_4, B_5, and H_5 are continuous. The variables are given in centimeters and the Young's modulus of the material is equal to 200 GPa. The volume of the beam, to be minimized, is given by

$$V(H_i, B_i) = 100 \sum_{i=1}^{5} H_i B_i$$

subject to

$$g_i(H_i, B_i) = \sigma_i \leq 14000\text{N/cm}^2 \quad i = 1, \ldots, 5$$
$$g_{i+5}(H_i, B_i) = H_i/B_i \leq 20 \quad i = 1, \ldots, 5$$
$$g_{11}(H_i, B_i) = \delta \leq 2.7\text{cm}$$

where δ is the tip deflection of the beam in the vertical direction. The Table IX presents some results found in the

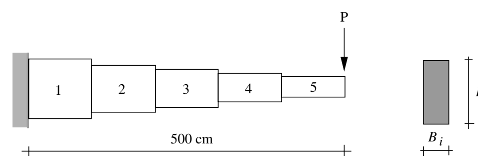

Fig. 5. The Cantilever Beam

literature and those found by using the algorithms proposed in this paper. An extended set of results for this problem can be found in [35]. The number of function evaluations was set equal to 35,000 for all experiments except in the Ref. [40] that used 10,000 function evaluations at each three levels of their GAOS algorithm. The AIS-GA without clearing found a better solution (65559.6) than the AIS-GA with clearing in this example. However, the GA proposed in [35] reaches a better result equal to 64698.6. The Table X shows the details of the final best solutions.

TABLE IX

VOLUME FOUND FOR THE CANTILEVER BEAM DESIGN

	nfe	Best	Average	Worst
Ref. [40]	10,000	64815	n.a.	n.a.
Ref. [35]	35,000	**64698.56**	**68107.046**	**73931.359**
AIS-GA	35,000	65559.60	70857.12	77272.78
AIS-GA^C	35,000	66533.47	71821.69	76852.86

F. The Ten-Bar Truss design

This is the well known test problem corresponding to the weight minimization of the ten-bar truss shown in the Figure 6. The constraints involve the stress in each member and the displacements at the nodes. The design variables are the cross-sectional areas of the bars (A_i, $i = 1, 10$). The allowable stress is limited to ± 25ksi and the displacements are limited to 2 in, in the x and y directions. The density of the material is 0.1 lb/in^3, Young's modulus is E = 10^4

TABLE X

VALUES FOUND FOR THE CANTILEVER BEAM DESIGN.

Var.	Ref. [40]	Ref. [35]	AIS-GA	AIS-GA^C
B_1	3	3	3	3
B_2	3.1	3.1	3.1	3.1
B_3	2.6	2.6	2.8	2.6
B_4	2.300	2.2894	2.2347884	2.3107138
B_5	1.800	1.7931	2.0038407	2.2254148
H_1	60	60	60	60
H_2	55	55	55	60
H_3	50	50	50	50
H_4	45.50	45.6256	44.39452	43.18571
H_5	35.00	34.5931	32.878708	31.250282
g_1	13888.89	13888.89	13888.889	13888.889
g_2	12796.59	12796.59	12796.588	10752.688
g_3	13846.15	13846.15	12857.143	13846.154
g_4	12600.87	12589.61	13622.479	13922.748
g_5	13605.44	13980.98	13849.324	13803.919
g_6	20.00	20.00	20.0	20.0
g_7	17.74	17.74	17.741936	19.35484
g_8	19.23	19.23	17.857143	19.23077
g_9	19.7826	19.9289	19.8652	18.689339
g_{10}	19.4444	19.2919	16.407845	14.042453
g_{11}	2.6960	2.6999	2.6999998	2.601907
V	64815	**64698.56**	65559.6	66533.47
nfe	10,000	35,000	35,000	35,000

ksi, and vertical downward loads of 100 kips are applied at nodes 2 and 4. Two cases are analyzed: discrete and

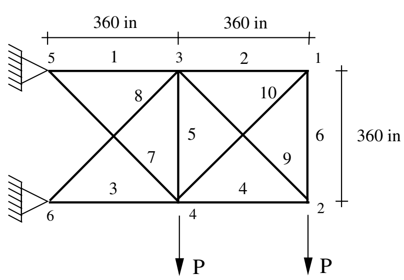

Fig. 6. The Ten-Bar Truss

continuous variables. For the discrete case the values of the cross-sectional areas (in^2) are chosen from the set $\mathcal{S}$ with 32 options: 1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.93, 3.13, 3.38, 3.47, 3.55, 3.63, 3.88, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.97, 11.50, 13.50, 14.20, 15.50, 16.90, 18.80, 19.90, 22.00, 26.50, 30.00, 33.50. For the continuous case the minimum cross sectional area is equal to 0.1 in^2. The Table XI presents the values found for the final weight of the Ten-bar Truss design considering the discrete case and using 90,000 function evaluations. The best solutions (5490.738) was found in the reference [35]. The Table XIII presents the values for the Ten-bar Truss design for the continuous case where the AIS-GA found the best solution equal to 5062.675 considering 280,000 objective function evaluations. An extended discussion of results for this problem can be found in [35]. The Tables XII and XIV show the final values of the design variables for the discrete and continuous cases, respectively.

TABLE XI

VALUES OF WEIGHT FOR THE TEN-BAR TRUSS – DISCRETE CASE

	Best	Average	Worst
Ref. [35]	**5490.74**	**5545.48**	**5567.84**
AIS-GA	5539.243	5754.969	6790.8936
AIS-GA^c	5528.087	5723.7837	6239.992

TABLE XII

VALUES FOUND FOR THE TEN-BAR TRUSS DESIGN – DISCRETE CASE.

Var.	Ref. [35]	AIS-GA	AIS-GA^C
1	33.50	33.50	33.5
2	1.62	1.8	1.62
3	22.90	26.5	22.0
4	14.20	15.50	14.2
5	1.62	1.62	1.62
6	1.62	2.13	1.62
7	7.97	7.97	5.74
8	22.90	19.9	26.5
9	22.00	22.0	22.0
10	1.62	1.62	1.62
W	**5490.738**	5539.243	5528.087
nfe	90,000	90,000	90,000

VI. CONCLUSIONS

A genetic algorithm hybridized with an artificial immune system was proposed and tested in a well known set of mixed constrained optimization problems in mechanical engineering. A comparison with some alternative approaches was performed and the AIS-GA provided competitive results in all experiments performed. One can observe that the proposed algorithm performed very well in problems presenting continuous design variables, reaches good results in problems with mixed design variables and, finally, shows a decrease in performance for problems with discrete design variables. Overall, the best performance among AIS inspired procedures was delivered by the AIS-GA hybrid proposed here. The introduction of a clearing procedure improved the quality of the results in almost all problems tested. The proposed hybrid can also be applied to other engineering problems and should be tested in larger mixed constrained optimization problems in mechanical engineering.

ACKNOWLEDGMENT

The authors acknowledge the support received from CNPq (grants 302299/2003-3 and 154674/2006-0).

REFERENCES

[1] M. Shoenauer and Z. Michalewicz, "Evolutionary computation at the edge of feasibility," in *Parallel Problem Solving from Nature - PPSN*

TABLE XIII

VALUES FOUND FOR THE FINAL WEIGHT OF THE TEN-BAR TRUSS DESIGN – CONTINUOUS CASE

	Best	Average	Worst
Ref. [35]	5069.09	5091.43	5117.39
AIS-GA	**5062.675**	**5075.5513**	**5094.8867**
AIS-GA^C	5064.669	5082.5156	5113.217

TABLE XIV

VALUES FOUND OF RESULTS OF TEN-BAR TRUSS DESIGN – CONTINUOUS CASE.

Var.	Ref. [35]	AIS-GA	AIS-GA^C
1	29.22568	30.162525	29.781208
2	0.10000	0.10003946	0.100310035
3	24.18212	22.81192	22.551401
4	14.94714	15.871827	15.504622
5	0.10000	0.10000233	0.10002254
6	0.39463	0.5149511	0.5237749
7	7.49579	7.505953	7.52854
8	21.92486	21.264076	21.15708
9	21.29088	21.383036	22.21351
10	0.10000	0.10000795	0.10018318
W	5069.086	**5062.675**	5064.669
nfe	280,000	280,000	280,000

IV, H.-M. V. W. E. I. Rechenberg and H.-P. Schwefel, Eds., vol. 1141. Berlin: Springer-Verlag, 1996, pp. 245–254, lNCS.

[2] S. Koziel and Z. Michalewicz, "Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization," *Evolutionary Computation*, vol. 7, no. 1, pp. 19–44, 1999.

[3] 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*, 1994, pp. 548–553.

[4] H. Adeli and N.-T. Cheng, "Augmented Lagrangian Genetic Algorithm for Structural Optimization," *Journal of Aerospace Engineering*, vol. 7, no. 1, pp. 104–118, January 1994.

[5] H. J. C. Barbosa, "A coevolutionary genetic algorithm for constrained optimization problems," in *Proc. of the Congress on Evolutionary Computation*, Washington, DC, USA, 1999, pp. 1605–1611.

[6] P. Surry and N. Radcliffe, "The COMOGA Method: Constrained Optimisation by Multiobjective Genetic Algorithms," *Control and Cybernetics*, vol. 26, no. 3, pp. 391–412, 1997.

[7] C. A. C. Coello and E. M. Montes, "Constraint-handling genetic algorithms through the use of dominance-based tournament selection," *Advanced Engineering Informatics*, no. 16, pp. 193–203, 2002.

[8] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, September 2000.

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

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

[11] H. H. S.-Y. Lai and X. Qi, "Constrained optimization via genetic algorithms," *Simulation*, vol. 62, no. 4, pp. 242–254, 1994.

[12] J. Joines and C. R. Houck, "On the use of non-stationary penalty methods to solve nonlinear constrained optimization problems with GAs," in *Proc. of 1994 IEEE Conf. on Evolutionary Computation*, D. Fogel and Z. Michalewicz, Eds., 1994, pp. 579–585.

[13] R. L. Riche, C. Knopf-Lenoir, and R. Haftka, "A segregated genetic algorithm for constrained structural optimization," in *Proc. of the Sixth Int. Conf. on Genetic Algorithms*, L. Eshelman, Ed., Pittsburgh, PA., July 1995, pp. 558–565.

[14] D. C. A. Smith and D. Tate, "Adaptive penalty methods for genetic optimization of constrained combinatorial problems," *INFORMS Journal on Computing*, vol. 6, no. 2, pp. 173–182, 1996.

[15] H. J. C. Barbosa and A. C. C. Lemonge, "An adaptive penalty scheme in genetic algorithms for constrained optimization problems," in *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*. New York: Morgan Kaufmann Publishers, 9-13 July 2002, pp. 287–294.

[16] ——, "An adaptive penalty scheme for steady-state genetic algorithms," pp. 718–729, July 2003.

[17] Z. Michalewicz and M. Shoenauer, "Evolutionary algorithms for constrained parameter optimization problems," *Evolutionary Computation*, vol. 4, no. 1, pp. 1–32, 1996.

[18] 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*, Alaska, May 4-9 1998, pp. 810–815.

[19] J.-H. Kim and H. Myung, "Evolutionary programming techniques for constrained optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 2, no. 1, pp. 129–140, 1997.

[20] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2-4, pp. 311–338, June 2000.

[21] S. B. Hamida and M. Shoenauer, "An adaptive algorithm for constrained optimization problems," in *Parallel Problem Solving from Nature - PPSN VI*, vol. 1917. Berlin: Springer-Verlag, 2000, pp. 529–538, lecture Notes in Computer Science.

[22] J. Wright and R. Farmani, "Genetic algorithms: A fitness formulation for constrained minimization." in *Proc. of the Genetic and Evolutionary Computation Conference – GECCO 2001*. San Francisco, CA.: Morgan Kaufmann., 2001, pp. 725–732.

[23] P. Hajela and J. Lee, "Constrained genetic search via schema adaptation. an immune network solution," in *1st World Congress of Stuctural and Multidisciplinary Optimization*. Goslar, Germany: Pergamon Press, 1995, pp. 915–920.

[24] ——, "Constrained genetic search via schema adaptation. An immune network solution," *Structural Optimization*, vol. 12, pp. 11–15, 1996.

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

[26] J. S. Yoo and P. Hajela, "Immune network simulations in multicriterion design," *Structural Optimization*, vol. 18, pp. 85–94, 1999.

[27] C. A. C. Coello and N. C. Cortés, "Hybridizing a genetic algorithm with an artificial immune system for global optimization," *Engineering Optimization*, vol. 36, no. 5, pp. 607–634, October 2004.

[28] N. C. Cortés, D. Trejo-Pérez, and C. A. C. Coello, "Handling constraints in global optimization using an artificial immune system." in *ICARIS*, ser. Lecture Notes in Computer Science. Banff, Canada: Springer, 2005, vol. 3627, pp. 234–247.

[29] L. N. de Castro and J. Timmis, "An artificial immune network for multimodal function optimization," in *Proc. of the 2002 IEEE World Congress on Computational Intelligence*, vol. I, Honolulu, Hawaii, USA, May 2002, pp. 669–674.

[30] L. N. de Castro and F. J. V. Zuben, "Learning and optimization using the clonal selection principle," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 3, pp. 239–251, 2002.

[31] H. S. Bernardino, H. J. C. Barbosa, and A. C. C. Lemonge, "Constraint handling in genetic algorithms via artificial immune systems," in *Late Breaking Paper, Genetic And Evolutionary Computation Conference, GECCO-2006*, Seattle, WA, USA, 2006.

[32] A. Petrowski., "A clearing procedure as a niching method for genetic algorithms." in *Proc. Third IEEE International Conference on Evolutionary Computation*, 1996, pp. pages 798–803.

[33] G. Singh and K. Deb, "Comparison of multi-modal optimization algorithms based on evolutionary algorithms," in *Genetic And Evolutionary Computation Conference, GECCO-2006*, Seattle, WA, USA, 2006.

[34] M. Efrén, C. A. C. Coello, and L. Ricardo, "Engineering optimzation using a simple evolutionary algorithm," in *15th Intl. Conf. on Tools with Art. Intelligence – ICTAI'2003*, CA, USA, 2003, pp. 149–156.

[35] A. C. C. Lemonge and H. J. C. Barbosa, "An adaptive penalty scheme for genetic algorithms in structural optimization," *Int. Journal for Numerical Methods in Engineering*, vol. 59, no. 5, pp. 703–736, 2004.

[36] E. Sandgren, "Nonlinear integer and discrete programming in mechanical design." in *Proc. of the ASME Design Technology Conference*, Kissimee, FL, 1988, pp. 95–105.

[37] B. Kannan and S. Kramer, "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design." *Journal of Mechanical Design*, no. 116, pp. 405–411, 1995.

[38] K. Deb, "GeneAS: A robust optimal design technique for mechanical component design," in *Evolutionary Algorithms in Engineering Applications*. Springer-Verlarg, Berlin, 1997, pp. 497–514.

[39] C. A. C. Coello, "Use of a Self-Adaptive Penalty Approach for Engineering Optimization Problems," *Computers in Industry*, vol. 41, no. 2, pp. 113–127, January 2000.

[40] F. Erbatur, O. Hasançebi, I. Tütüncü, and H. Kilç, "Optimal design of planar and space structures with genetic algorithms," *Computers & Structures*, vol. 75, pp. 209–224, 2000.