Asociación Argentina de Mecánica Computacional

Mecánica Computacional Vol XXIX, págs. 9287-9303 (artículo completo)
Eduardo Dvorkin, Marcela Goldschmit, Mario Storti (Eds.)
Buenos Aires, Argentina, 15-18 Noviembre 2010

CONSTRAINED OPTIMIZATION PROBLEMS IN MECHANICAL ENGINEERING DESIGN USING A REAL-CODED STEADY-STATE GENETIC ALGORITHM

Afonso C.C. Lemonge[a], Helio J.C. Barbosa[b], Carlos C.H. Borges[c] and Francilene B.S. Silva[d]

[a]*Departamento de Mecânica Aplicada e Computacional, Universidade Federal de Juiz de Fora, Campus Universitário, Bairro Martelos, 36036 330 Juiz de Fora MG, Brazil, afonso.lemonge@ufjf.edu.br, http://www.mmc.ufjf.br*

[b]*Laboratório Nacional de Computação Científica, Av. Getúlio Vargas 333, 25651 075 Petrópolis RJ, Brazil, hcbm@lncc.br, http://www.lncc.br/~hcbm*

[c]*Departamento de Ciência da Computação, Universidade Federal de Juiz de Fora, Campus Universitário, Bairro Martelos, 36036 330 Juiz de Fora MG, Brazil, cchb@lncc.br, http://www.mmc.ufjf.br*

[d]*Mestrado em Modelagem Computacional, Universidade Federal de Juiz de Fora, Campus Universitário, Bairro Martelos, 36036 330 Juiz de Fora MG, Brazil, francilene@ice.ufjf.br, http://www.mmc.ufjf.br*

Keywords: Constrained Optimization, Genetic Algorithm, Penalty Methods.

Abstract. A parameter-less adaptive penalty scheme for steady-state genetic algorithms applied to constrained optimization problems was proposed previously by two of co-authors of this paper. For each constraint, a penalty parameter is adaptively computed along the run according to information extracted from the current population such as the existence of feasible individuals and the level of violation of each constraint. In this paper the performance of this scheme is extended using test problems from the mechanical engineering design, also, largely tested in the evolutionary computation literature. Using real coding, rank-based selection and operators available in the literature very competitive results are obtained and those are compared with other techniques.

1 INTRODUCTION

Evolutionary algorithms (EAs) are weak search algorithms which can be directly applied to unconstrained optimization problems where one seeks for an element (vector of design variables), belonging to the search space, which minimizes (or maximizes) a real function. Such EA usually employs a fitness function closely related to this real function. To handle constrained optimization problems, it is usual to consider a penalty scheme coupled to the fitness function.

The straightforward application of EAs to constrained optimization problems (COPs) is not possible due to the additional requirement that a set of constraints must be satisfied. Several difficulties may arise: (i) the objective function may be undefined for some or all infeasible elements, (ii) the check for feasibility can be more expensive than the computation of the objective function value, and (iii) an informative measure of the degree of infeasibility of a given candidate solution is not easily defined. It is easy to see that even if both the objective function $f(x)$ and a measure of constraint violation $v(x)$ are defined for all $x \in S$ it is not possible to know in general which of two given infeasible solutions is closer to the optimum and thus should be operated upon or kept in the population.

The techniques for handling constraints within EAs can be classified either as *direct* (feasible or interior) Shoenauer and Michalewicz (1996), Koziel and Michalewicz (1999), Liepins and Potter (1991); Orvosh and Davis (1994), when only feasible elements in S are considered or exterior *indirect* that considers both feasible and infeasible elements during the search process Adeli and Cheng (1994), Barbosa (1999), Surry and Radcliffe (1997), Runarsson and Yao (2000), and van Kampen et al. (1996).

A parameter-less adaptive penalty scheme for steady-state genetic algorithms applied to constrained optimization problems was previously proposed in the literature by Barbosa and Lemonge Barbosa and Lemonge (2003) and used to find solutions in a suite of functions exhaustively tested as benchmark problems. For each constraint, a penalty parameter is adaptively computed along the run according to information extracted from the current population such as the existence of feasible individuals and the level of violation of each constraint. The idea is that the values of the penalty coefficients should be distributed in a way that those constraints which are more difficult to be satisfied should have a relatively higher penalty coefficient.

In this paper the performance of this scheme is extended using test problems from the mechanical engineering design, also, largely tested in the evolutionary computation literature. Using real coding, rank-based selection and operators available in the literature very competitive results are obtained and those are compared with other techniques.

In the next section the constrained optimization problem is defined. The adaptive penalty method for real-coded steady-state GA and is presented Section 4 presents experimental study with test-problems from the mechanical engineering design literature and the paper closes with some conclusions in Section 5.

2 CONSTRAINED OPTIMIZATION PROBLEM

A standard constrained optimization problem in R^n can be thought of as the minimization of the objective function $f(x)$, 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 $x_i^L \leq x_i \leq x_i^U$ and very often 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 the constraints are in fact a complex, often computationally expensive, *implicit* function of the design variables. 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. In the next section the Adaptive Penalty Method for the real-coded steady-state GA is described.

3 THE ADAPTIVE PENALTY METHOD FOR THE REAL-CODED STEADY-STATE GA

The Adaptive Penalty Method (APM) scheme presented in Barbosa and Lemonge (2002) adaptively sizes the penalty coefficient of each constraint using information from the population such as the average of the objective function and the level of violation of each constraint. The fitness function is written as:

$$F(x) = \begin{cases} f(x), & \text{if } x \text{ is feasible,} \\ \mathsf{h}(x) + \sum_{j=1}^{m} k_j v_j(x) & \text{otherwise} \end{cases} \tag{1}$$

where

$$\mathsf{h}(x) = \begin{cases} f(x), & \text{if } f(x) > \langle f(x) \rangle, \\ \langle f(x) \rangle & \text{otherwise} \end{cases} \tag{2}$$

and $\langle f(x) \rangle$ is the average of the objective function values in the current population. The penalty parameter is defined at each *generation* by:

$$k_j = |\langle f(x) \rangle| \frac{\langle v_j(x) \rangle}{\sum_{l=1}^{m} [\langle v_l(x) \rangle]^2} \tag{3}$$

and $\langle v_l(x) \rangle$ is the violation of the l-th constraint averaged over the current population. The idea is that the values of the penalty coefficients should be distributed in a way that those constraints which are more difficult to be satisfied should have a relatively higher penalty coefficient.

The straightforward extension of the penalty procedure proposed in Barbosa and Lemonge (2002) to the steady-state case would be to periodically update the penalty coefficients and the fitness function values for the population.

Further modifications were then proposed for the steady-state version of the penalty scheme. The fitness function is still computed according to (1). However, h and the penalty coefficients are redefined respectively as

$$\mathsf{h} = \begin{cases} f(x_{worst}) & \text{if there is no feasible element in the population,} \\ f(x_{bestfeasible}) & \text{otherwise} \end{cases} \tag{4}$$

$$k_j = \mathtt{h} \frac{\langle v_j(x) \rangle}{\sum_{l=1}^{m} [\langle v_l(x) \rangle]^2} \tag{5}$$

Also, every time a better feasible element is found (or the number of new elements inserted into the population reaches a certain level) h is redefined and all fitness values are recomputed using the updated penalty coefficients. The updating of each penalty coefficient is performed in such a way that no reduction in its value is allowed. For convenience one should keep the objective function value and all constraint violations for each individual in the population. The fitness function value is then computed using (4), (5), and (1).

It is clear from the definition of $\mathtt{h}$ in (4) that if no feasible element is present in the population one is actually minimizing a measure of the distance of the individuals to the feasible set since the actual value of the objective function is not taken into account. However, when a feasible element is found then it immediately enters the population since, after updating all fitness values using (4), (5), and (1), it becomes the element with the best fitness value.

A pseudo-code for the proposed adaptive penalty scheme for a steady-state GA can be written as shown in Figure 1. Numerical study are then presented in the following section.

```
procedure RCSS
    Initialize population
    Compute objective function and constraint violation values
    if there is no feasible element then
        h ← worst objective function value
    else
        h ← objective function value of best feasible individual
    end if
    Compute penalty coefficients
    Compute fitness values
    ninser = 0
    for i = 1:maxeval do
        Select operator
        Select parent(s)
        Generate offspring
        Evaluate offspring
        Keep best offspring
        if offspring is the new best feasible element then
            update penalty coefficients and fitness values
            ninser = 0
        end if
        if offspring is better than the worst in the population then
            worst is removed
            offspring is inserted
            ninser = ninser + 1
        end if
        if ninser/popsize ≥ r then
            update penalty coefficients and fitness values
            ninser = 0
        end if
    end for
end procedure
```

Figure 1: Pseudo-code for the steady-state GA with adaptive penalty scheme. (`ninser` is a counter for the number of offspring inserted in the population, `popsize` is the population size, `maxeval` is the maximum number of function evaluation and `r` is a fixed constant that was set to 3 in all cases).

4 EXPERIMENTAL STUDY

Six mechanical engineering optimization problems are used to assess the performance of the proposed algorithm when compared with alternative constraint handling techniques: Artificial Immune Systems hybridized with a GA (AIS-GA and AIS-GA^C) from Bernardino et al. (2007); AIS-GA^H from Bernardino et al. (2008); Adaptive Penalty Method (APM^{bc}), presented in Bernardino et al. (2008) –here the superscript bc is added in order to inform the use of a "binary code"–; The Stochastic Ranking technique (SR) proposed by Runarsson and Yao (2000) and used within a generational GA with a binary code presented in Bernardino et al. (2008). Also, some algorithms are used in the comparisons: The Evolution Strategy (ES-Coello) proposed in M.M. Efrén and Ricardo (2003); The AIS-Coello presented in M.-Montes et al. (2003); The GA (GAOS-Erbatur) proposed in Erbatur et al. (2000). The results of the present work are defined by APM^{rc} where the notation rc means the use of a "real code"

The APM^{bc} and the SR method presented in Bernardino et al. (2008) used a population size of 100 individuals, and a mutation rate equal to 0.004. Also, elitism is implemented: a copy of the best individual remains in the next generation. For the AIS-GA^H the population size was set to 50, the mutation rate was set to 0.01, and elitism was implemented (the two best individuals are copied to the next generation). Also, each antibody generates only one clone, and the maximum mutation rate of the AIS was set to 0.03. All techniques use a binary Gray code with 25 bits for each continuous variable, and a crossover probability equal to 0.9.

The simple real-coded steady-state GA (APM^{rc}), with a linear ranking selection scheme was implemented using: (i) random mutation (which modifies a randomly chosen variable of the selected parent to a random value uniformly distributed between the lower and upper bounds of the corresponding variable), (ii) non-uniform mutation (as proposed by Michalewicz Michalewicz (1992)), (iii) Muhlenbein's mutation (as described in Muhlenbein et al. (1991)), (iv) multi-parent discrete crossover (which generates an offspring by randomly taking each allele from one of the n_p selected parents), and (v) Deb's SBX crossover as described inDeb and Agrawal (1995).

No parameter tuning was attempted. The same probability of application (namely 0.2) was assigned to all operators above, n_p was set to 4, and η was set to 2 in SBX. This set of values was applied to *all* test-problems, solved by APM^{rc}, in order to demonstrate the robustness of the procedure. The values of the best, median, average, standard deviation, the worst and the number of runs that reached feasible solutions are presented for each experiment corresponding to a number of function evaluations set for each of them.

4.1 The Tension/Compression Spring Design

The objective is to minimize the volume V of a coil spring under a constant tension/compression load. The design variables are the number of active coils of the spring ($N = x_1 \in [2, 15]$), the winding diameter ($D = x_2 \in [0.25, 1.3]$), and the wire diameter ($d = x_3 \in [0.05, 2]$). The volume and the mechanical constraints are given by:

$$V(x) = (x_1 + 2)x_2x_3^2$$
$$g_1(x) = 1 - \frac{x_2^3 x_1}{71785x_3^4} \leq 0$$
$$g_2(x) = \frac{4x_2^2 - x_3x_2}{12566(x_2x_3^3 - x_3^4)} + \frac{1}{5108x_3^2} - 1 \leq 0$$
$$g_3(x) = 1 - \frac{140.45x_3}{x_2^2x_1} \leq 0$$
$$g_4(x) = \frac{x_2 + x_3}{1.5} - 1 \leq 0$$

where

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

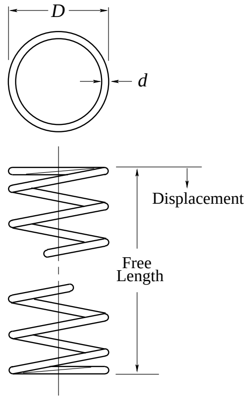

Figure 2: The Tension/Compression Spring

The number of function evaluations was set to 36,000 (200 individuals in the population). A comparison of results is provided in the Table 1 where the best result is found by the AIS-GA^C and AIS-GA^H (Bernardino et al. (2007),Bernardino et al. (2008)) with a final volume equal to 0.012666. The Table 2 shows the values found for the design variables corresponding to the best solutions which are all feasible.

4.2 The Speed Reducer design

The objective is to minimize the weight W of a speed reducer. The design variables are the face width ($b = x_1 \in [2.6, 3.6]$), the module of teeth ($m = x_2 \in [0.7, 0.8]$), the number of teeth on pinion ($n = x_3 \in [17, 28]$), the length of the shaft 1 between the bearings ($l_1 = x_4 \in [7.3, 8.3]$), the length of the shaft 2 between the bearings ($l_2 = x_5 \in [7.8, 8.3]$), the diameter of the shaft 1 ($d_1 = x_6 \in [2.9, 3.9]$), and, finally, the diameter of the shaft 2 ($d_2 = x_7$). The variable x_3 is integer and all the others are continuous. The weight and the mechanical constraints are

	Best	Median	Average	St.Dev	Worst	fr
AIS-GA	0.012668	–	0.013481	–	0.016155	–
AIS-GA^C	**0.012666**	–	**0.012974**	–	**0.013880**	–
AIS-GA^H	**0.012666**	0.012892	0.013131	$\mathbf{6.28E-4}$	0.015318	**50**
APM^{bc}	0.012684	0.013575	0.014022	$1.47E-3$	0.017794	**50**
SR	0.012679	0.013655	0.013993	$1.27E-3$	0.017796	**50**
APM^{rc}	0.012679	**0.012733**	0.014466	$1.09E-2$	0.089992	**50**

Table 1: Values found for Tension/Compression Spring design.

	AIS-GA	AIS-GA^C	AIS-GA^H	APM^{bc}	SR	APM^{rc}
x_1	11.852177	11.329555	11.6611924	12.070748	11.375795	11.23705
x_2	0.347475	0.356032	0.3505298	0.344304	0.355485	0.357848
x_3	0.051302	0.051661	0.0514305	0.051168	0.051638	0.517359
V	0.012668	**0.012666**	**0.012666**	0.0126838	0.012679	0.012679

Table 2: Design variables found for the best solutions for the Tension/Compression Spring design

given by

$$\begin{aligned}
&W = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right)\\
&-1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right)\\
&+0.7854\left(x_4x_6^2 + x_5x_7^2\right)\\
&g_1(x) = 27x_1^{-1}x_2^{-2}x_3^{-1} \leq 1\\
&g_2(x) = 397.5x_1^{-1}x_2^{-2}x_3^{-2} \leq 1\\
&g_3(x) = 1.93x_2^{-1}x_3^{-1}x_4^3x_6^{-4} \leq 1\\
&g_4(x) = 1.93x_2^{-1}x_3^{-1}x_5^3x_7^{-4} \leq 1\\
&g_5(x) = \frac{1}{0.1x_6^3}\left[\left(\frac{745x_4}{x_2x_3}\right)^2 + \{16.9\}10^6\right]^{0.5} \leq 1100\\
&g_6(x) = \frac{1}{0.1x_7^3}\left[\left(\frac{745x_5}{x_2x_3}\right)^2 + (157.5)\,10^6\right]^{0.5} \leq 850\\
&g_7(x) = x_2x_3 \leq 40 \qquad g_8(x) = x_1/x_2 \geq 5\\
&g_9(x) = x_1/x_2 \leq 12 \quad g_{10}(x) = (1.5x_6 + 1.9)\,x_4^{-1} \leq 1\\
&g_{11}(x) = (1.1x_7 + 1.9)\,x_5^{-1} \leq 1
\end{aligned}$$

The Table 3 presents a comparison of the results found by the proposed algorithm and others from the literature. The number of function evaluations was set equal to 36,000. (200 individuals in the population). In this case one can observe that all techniques, except the ES-Coello, found essentially the same optimal design. The best value was found by the APM^{bc} and APM^{rc} (2996.3482). The Table 4 presents the final values of the design variables (all solutions are feasible).

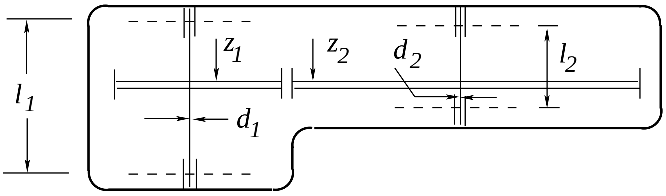

Figure 3: The Speed Reducer

	Best	Median	Average	St.Dev	Worst	fr
ES-Coello	3025.0051	–	3088.7778	–	3078.5918	–
AIS-GA *	2996.3494	2996.356	2996.3643	$4.35E-3$	2996.6277	**50**
AIS-GA^{C} *	2996.3484	2996.3484	**2996.3484**	$\mathbf{1.46E-6}$	**2996.3486**	**50**
AIS-GA^{H}	2996.3483	2996.3495	2996.3501	$7.45E-3$	2996.3599	**50**
APM^{bc}	**2996.3482**	**2996.3482**	3033.8807	$1.10E+2$	3459.0948	19
SR	2996.3483	2996.3488	2996.3491	$1.01E-3$	2996.3535	**50**
APM^{rc}	**2996.3482**	**2996.3482**	2997.4728	$7.87E+0$	3051.4556	49

Table 3: Values found for the Speed Reducer design

	ES-Coello	AIS-GA *	AIS-GA^{C} *	AIS-GA^{H}	APM^{bc}	SR	APM^{rc}
x_1	3.506163	3.500001	3.500000	3.500001	3.500000	3.500000	3.500000
x_2	0.700831	0.700000	0.700000	0.700000	0.700000	0.700000	0.700000
x_3	17	17	17	17	17	17	17
x_4	7.460181	7.300019	7.300001	7.300008	7.300000	7.300001	7.300000
x_5	7.962143	7.800013	7.800000	7.800001	7.800000	7.800001	7.800000
x_6	3.362900	3.350215	3.350215	3.350215	3.350215	3.350215	3.350215
x_7	5.308949	5.286684	5.286684	5.286683	5.286683	5.286683	5.286683
W	3025.0051	2996.3494	2996.3484	2996.3483	**2996.3482**	2996.3483	**2996.3482**

Table 4: Design variables found for the best solutions for the Speed Reducer design

4.3 The Welded Beam design

The objective is to minimize the cost $C(h, l, t, b)$ of the beam where $h \in [0.125, 10]$, and $0.1 \leq l, t, b \leq 10$. The objective and constraints read Deb (2000)

$$
\begin{aligned}
&C(h, l, t, b) = 1.10471h^2 l + 0.04811tb(14.0 + l) \\
&g_1(\tau) = 13,600 - \sqrt{(\tau')^2 + (\tau'')^2 + l\tau'\tau''/\alpha} \geq 0 \\
&g_2(\sigma) = 30,000 - 504000/(t^2 b) \geq 0 \\
&g_3(b, h) = b - h \geq 0 \qquad g_4(P_c) = P_c - 6,000 \geq 0 \\
&g_5(\delta) = 0.25 - 2.1952/(t^3 b) \geq 0 \\
&\tau' = \frac{6000}{\sqrt{2}hl} \qquad \alpha = \sqrt{0.25(l^2 + (h+t)^2)} \\
&P_c = 64746.022(1 - 0.0282346t)tb^3 \\
&\tau'' = \frac{6000(14 + 0.5l)\alpha}{2(0.707hl(l^2/12 + 0.25(h+t)^2))}
\end{aligned}
$$

$$
\begin{aligned}
&C(h, l, t, b) = 1.10471h^2 l + 0.04811tb(14.0 + l) \\
&g_1(\tau) = 13,600 - \tau \geq 0 \qquad g_2(\sigma) = 30,000 - \sigma \geq 0 \\
&g_3(b, h) = b - h \geq 0 \qquad g_4(P_c) = P_c - 6,000 \geq 0 \\
&g_5(\delta) = 0.25 - \delta \geq 0
\end{aligned}
$$

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

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

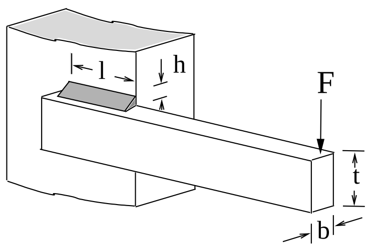

Figure 4: The Welded Beam

The Table 5 shows a comparison of results where the best value found (final cost equal to 2.38122) corresponds to the AIS-GA^C presented in the reference Bernardino et al. (2007). The Table 6 shows the design variables corresponding to the best solution found by each technique. All the solutions are feasible and the number of function evaluations was set to 320,000 (200 individuals in the population).

	Best	Median	Average	St.Dev	Worst	fr
AIS-GA	2.38125	–	2.59303	–	3.23815	–
AIS-GA^C	**2.38122**	–	**2.38992**	–	**2.41391**	–
AIS-GA^H	2.38335	**2.92121**	2.99298	**2.02E − 1**	4.05600	**50**
APM^{bc}	2.38144	3.27244	3.49560	$9.09E-1$	5.94803	**50**
SR	2.59610	4.21812	4.33259	1.29	10.1833	**50**
APM^{rc}	2.38124	2.67099	6.24497	$2.29E+1$	165.13681	50

Table 5: Values found for the cost of the Welded Beam design.

	AIS-GA	AIS-GA^C	AIS-GA^H	APM^{bc}	SR	APM^{rc}
h	0.2443243	0.2443857	0.2434673	0.2442419	0.2758192	0.244395
l	6.2201996	6.2183037	6.2507296	6.2231189	5.0052613	6.218086
t	8.2914640	8.2911650	8.2914724	8.2914718	8.6261101	8.291043
b	0.2443694	0.2443875	0.2443690	0.2443690	0.2758194	0.244395
Cost	2.381246	**2.38122**	2.38335	2.38144	2.59610	2.38124

Table 6: Design variables found for the best solutions for the Welded Beam design

4.4 The Pressure Vessel design

This problem Sandgren (1988); Kannan and Kramer (1995); Deb (1997); Coello Coello (2000) corresponds to the weight minimization of a cylindrical pressure vessel with two spherical heads. There are four design variables (in inches): 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 weight, to be minimized, and the constraints are given by:

$$
\begin{aligned}
&W(T_s, T_h, R, L) = 0,6224 T_s T_h R + \\
&+1.7781 T_h R^2 + 3.1661 T_s^2 L + 19.84 T_s^2 R \\
&g_1(T_s, R) = T_s - 0.0193 R \geq 0 \\
&g_2(T_h, R) = T_h - 0.00954 R \geq 0 \\
&g_3(R, L) = \pi R^2 L + 4/3 \pi R^3 - 1,296,000 \geq 0 \\
&g_4(L) = -L + 240 \geq 0
\end{aligned}
$$

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 7 provides a comparison of results obtained with different algorithms. All algorithms use 80,000 function evaluations, except AIS M.-Montes et al. (2003) which used 150,000. The APM^{rc} used 200 individuals in the population. The best solution was found by APM^{rc} and corresponds to a final weight equal to 6059.715. The Table 8 displays the final solutions which are all feasible.

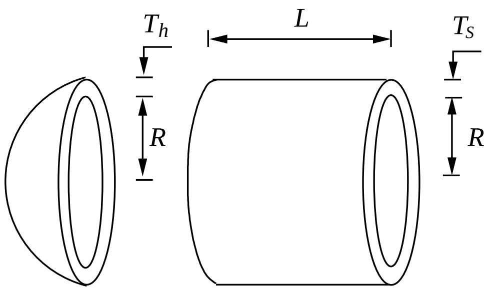

Figure 5: The Pressure Vessel.

	Best	Median	Average	St.Dev	Worst	fr	ne
AIS-Coello	6061.123	–	6734.085	–	7368.060	–	150,000
AIS-GA	6060.368	–	6743.872	–	7546.750	–	80,000
AIS-GA^C	6060.138	–	**6385.942**	–	**6845.496**	–	80,000
AIS-GA^h	6059.855	6426.710	6545.126	**1.24E + 2**	7388.160	**50**	80,000
APM^{bc}	6065.822	6434.435	6632.376	$5.15E+2$	8248.003	**50**	80,000
SR	6832.584	7073.107	7187.314	$2.67E+2$	8012.651	**50**	80,000
APM^{rc}	**6059.715**	**6288.529**	**6344.079**	$2.78E+2$	6928.386	49	80,000

Table 7: Values of the weight found for the Pressure Vessel design

	AIS-Coello	AIS-GA	AIS-GA^C	AIS-GA^H	APM^{bc}	SR	APM^{rc}
T_s	0.8125	0.8125	0.8125	0.8125	0.8125	1.1250	0.8125
T_h	0.4375	0.4375	0.4375	0.4375	0.4375	0.5625	0.4375
R	42.0870	42.0931	42.0950	42.0973	42.0492	58.1267	42.0984
L	176.7791	176.7031	176.6797	176.6509	177.2522	44.5941	176.6368
W	6061.1229	6060.3677	6060.138	**6059.8546**	6065.8217	6832.5836	**6059.715**

Table 8: Design variables found for the best solutions for the Pressure Vessel design

4.5 The Cantilever Beam design

This test problem Erbatur et al. (2000) corresponds to the minimization of the volume of a cantilever beam subject to the load $P = 50000$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. 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 and the constraints read:

$$
\begin{aligned}
&V(H_i, B_i) = 100 \sum_{i=1}^{5} H_i B_i \\
&g_i(H_i, B_i) = \sigma_i \leq 14000\text{N/cm}^2 \quad i = 1, \ldots, 5 \\
&g_{i+5}(H_i, B_i) = H_i/B_i \leq 20 \quad i = 1, \ldots, 5 \\
&g_{11}(H_i, B_i) = \delta \leq 2.7\text{cm}
\end{aligned}
$$

where δ is the tip deflection of the beam in the vertical direction.

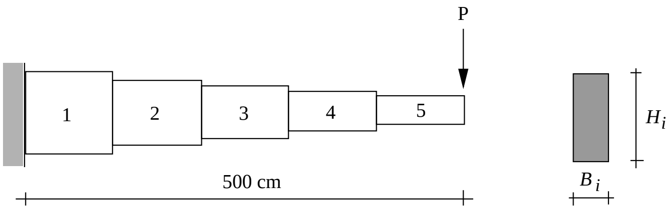

Figure 6: The Cantilever Beam

The Table 9 presents results found using different techniques. The number of function evaluations was set to 35,000 in all cases, except in the Ref. Erbatur et al. (2000) that used 10,000 function evaluations in each one of the three levels of the GAOS algorithm. The APM^{rc} used 350 individuals in the population. The SR technique produced the best solution with a final volume equal to 64599.6509715. The Table 10 shows the design variables values corresponding to the best solutions (all feasible) where "ne" corresponds to the number of function evaluations.

	Best	Median	Average	St.Dev	Worst	fr	ne
GAOS-Erbatur	64815	–	–	–	–	–	10000
AIS-GA	65559.60	–	**70857.12**	–	77272.78	–	35000
AIS-GA^C	66533.47	–	71821.69	–	**76852.86**	–	35000
AIS-GA^H	64834.70	74987.16	76004.24	$6.93E+3$	102981.06	**50**	35000
APM^{bc}	66030.05	79466.10	83524.21	$1.44E+4$	151458.17	**50**	35000
SR	**64599.65**	**70508.33**	71240.03	**3.90E + 3**	83968.45	47	35000
APM^{rc}	64647.82	76721.19	79804.77	$1.63E+4$	162089.24	49	35000

Table 9: Volume found for the Cantilever Beam design

	GAOS-Erbatur	AIS-GA	AIS-GA^C	AIS-GA^H	APM^{bc}	SR	APM^{rc}
B_1	3	3	3	3	3	3	3
B_2	3.1	3.1	3.1	3.1	3.1	3.1	3.1
B_3	2.6	2.8	2.6	2.6	2.6	2.6	2.6
B_4	2.3000	2.2348	2.3107	2.2947	2.2094	2.2837	2.2978
B_5	1.8000	2.0038	2.2254	1.8250	2.0944	1.7532	1.7574
H_1	60	60	60	60	60	60	60
H_2	55	55	60	55	60	55	55
H_3	50	50	50	50	50	50	50
H_4	45.5000	44.3945	43.1857	45.2153	44.0428	45.5507	45.5037
H_5	35.0000	32.878708	31.250282	35.1191	31.9867	35.0631	34.9492
V	64815	65559.6	66533.47	64834.70	66030.05	$\mathbf{64599.65}$	64647.82
ne	$10,000$	$35,000$	$35,000$	$35,000$	$35,000$	$35,000$	$35,000$

Table 10: Design variables found for the best solutions for the Cantilever Beam design

4.6 The Ten-Bar Truss design

This is a well known test problem corresponding to the weight minimization of a ten-bar truss structure. 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 $\pm$ 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 ksi, and vertical downward loads of 100 kips are applied at nodes 2 and 4.

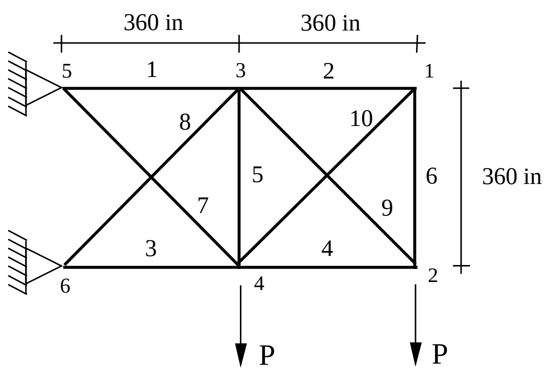

Figure 7: The ten-Bar Truss

Two cases are analyzed: discrete and 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 number of function evaluations considered were 90,000 and 280,000 for the discrete and continuous cases, respectively. For the APM^{rc} 300 and 400 individuals were used in the population for the discrete and continuous case, respectively.

The Table 11 presents the values found for the final weight in the discrete case. c APM^{cc} and AIS-GA^{H} produced the best solution (5490.738 lbs).

The Table 13 presents the values for the continuous case, where the APM^{rc} found the best solution (5060.99). The Tables 12 and 14 show the final values of the design variables for the discrete and continuous cases, respectively, which are all feasible.

	Best	Median	Average	St.Dev	Worst	fr
AIS-GA	5539.24	–	5754.97	–	6790.89	–
AIS-GA^{C}	5528.09	–	5723.78	–	6239.99	–
AIS-GA^{H}	**5490.74**	**5504.54**	**5513.90**	**2.56E + 1**	**5575.28**	**50**
APM^{bc}	**5490.74**	5558.74	5585.98	$1.48E + 2$	6443.23	**50**
SR	5491.72	5648.46	5664.21	$9.64E + 1$	6020.77	**50**
APM^{rc}	**5490.74**	5593.24	5607.49	$9.01E + 1$	5891.16	48

Table 11: Values of weight for the Ten-bar Truss – discrete case

	AIS-GA	AIS-GA^{C}	AIS-GA^{H}	APM^{bc}	SR	APM^{rc}
1	33.50	33.50	33.50	33.50	33.50	33.50
2	1.80	1.62	1.62	1.62	1.62	1.62
3	26.5	22.00	22.90	22.90	22.90	22.90
4	15.50	14.20	14.20	14.20	15.50	14.20
5	1.62	1.62	1.62	1.62	1.62	1.62
6	2.13	1.62	1.62	1.62	1.62	1.62
7	7.97	5.74	7.97	7.97	7.97	7.97
8	19.90	26.50	22.90	22.90	22.00	22.90
9	22.00	22.00	22.00	22.00	22.00	22.00
10	1.62	1.62	1.62	1.62	1.62	1.62
W	5539.24	5528.09	**5490.74**	**5490.74**	5491.72	**5490.74**

Table 12: Design variables found for the best solutions for the Ten-bar Truss design – discrete case

4.7 Discussion

The Table 15 shows the performance of the algorithms used in the comparisons in this paper. The Adaptive Penalty Method in the realcoded steady-state algorithm APM^{rc} reached very competitive values, particularly, in finding the best solutions. In the first column of this table this technique appears four times in seven possible problems (seven test-problems), i.e., the APM^{rc} presented a rate of 57.24 % of success. Although, this rate not appear significantly in the median, average and worst values, it is possible to observe, from the tables throughout this paper, that these values found by APM^{rc} shown to be very competitive. Probably, changing the GA parameters as the number of runs, population size and type of operators one can reach competitive and better values not only for the best but for average, median and worst solutions, but no parameter tuning was attempted in this way. The differences, in all metrics, sometimes occurred in the significant numeral after the floating point.

	Best	Median	Average	St.Dev	Worst	fr
AIS-GA	5062.67	–	5075.55	–	5094.89	–
AIS-GA^C	5064.67	–	5082.52	–	5113.22	–
AIS-GA^H	5061.16	**5064.36**	**5068.85**	**7.78**	**5084.56**	**50**
APM^{bc}	5062.12	5070.54	5133.22	$2.48E+2$	6430.55	**50**
SR	5061.71	5079.53	5077.67	$1.01E+1$	5101.17	**50**
APM^{rc}	**5060.99**	5075.86	5109.83	$2.20E+2$	6629.79	**50**

Table 13: Values found for the final weight of the Ten-bar Truss design – continuous case

	AIS-GA	AIS-GA^C	AIS-GA^H	APM^{rc}	SR	APM^{rc}
1	30.16252	29.78121	30.52684	30.95080	30.01400	30.41463
2	0.10004	0.10031	0.10000	0.10000	0.10000	0.10000
3	22.81192	22.55140	22.91574	22.92083	26.14460	23.18510
4	15.87183	15.50462	15.48294	15.55024	15.29260	15.17496
5	0.10000	0.10002	0.10000	0.10000	0.10000	0.10000
6	0.51495	0.52377	0.54620	0.60959	0.55610	0.54325
7	7.50595	7.52854	7.47594	7.46973	7.43980	7.44463
8	21.26408	21.15708	21.01566	20.83562	21.00560	20.97122
9	21.38304	22.21351	21.55362	21.35644	21.93900	21.73486
10	0.10001	0.10018	0.10000	0.10000	0.10000	0.10000
W	5062.67	5064.67	5061.16	5062.12	5061.71	**5060.99**

Table 14: Design variables found for the best solutions for the Ten-bar Truss design – Continuous case

	Best	Median	Average	Worst
T/C. Spring	AIS-GA^C	**APM**^{rc}	AIS-GA^C	AIS-GA^C
S. Reducer	APM^{bc}/**APM**^{rc}	APM^{bc}	AIS-GA^C	AIS-GA^C
W. Beam	AIS-GA^C	AIS-GA^H	AIS-GA^C	AIS-GA^C
P. Vessel	**APM**^{rc}	**APM**^{rc}	**APM**^{rc}	AIS-GA^C
C. Beam	SR	SR	AIS-GA	AIS-GA^C
10-bar (dis)	AIS-GA/APM^{bc}/**APM**^{rc}	AIS-GA^H	AIS-GA^H	AIS-GA^H
10-bar (con)	**APM**^{rc}	AIS-GA^H	AIS-GA^H	AIS-GA^H

Table 15: Best performing technique in each mechanical engineering problem

5 CONCLUSIONS

An adaptive parameter-less penalty scheme has been proposed in Barbosa and Lemonge (2003) in order to tackle constrained optimization problems. Its main feature, besides being adaptive and not requiring any parameter, is to automatically define a different penalty coefficient for each constraint. The algorithm, as expected, performed very well in problems of optimization from the mechanical engineering design. The problems discussed present continuous, discrete, and mixed design variables. Besides, the algorithm used produced competitive results compared with other techniques found in the literature in all problems tested so far. The next studies will discuss larger structural optimization problems.

5.1 Acknowledgments

The authors would like to thank the support from CNPq (grants 308317/2009-2 and 301527/2008-3), FAPEMIG (TEC 425/09), and FAPERJ (grants E-26/102.825/2008 and E-26/100.308/2010).

REFERENCES

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

Barbosa H. A coevolutionary genetic algorithm for constrained optimization problems. In *Proc. of the Congress on Evolutionary Computation*, pages 1605–1611. Washington, DC, USA, 1999.

Barbosa H. and Lemonge A. An adaptive penalty scheme in genetic algorithms for constrained optimization problems. In *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conf.*, pages 287–294. Morgan Kaufmann Publishers, New York, 2002.

Barbosa H. and Lemonge A. An adaptive penalty scheme for steady-state genetic algorithms. 2723:718–729, 2003.

Bernardino H., Barbosa H., and Lemonge A. A hybrid genetic algorithm for constrained optimization problems in mechanical engineering. In *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, pages 646–653. Singapore:IEEE Press, 2007.

Bernardino H., Barbosa H., Lemonge A., and Fonseca L. A new hybrid AIS-GA for constrained optimization problems in mechanical engineering. In *Proceedings of the 2008 IEEE Congress on Evolutionary Computation*, pages 1455–1462. Hong Kong:IEEE Press, 2008.

Coello Coello C.A. Use of a Self-Adaptive Penalty Approach for Engineering Optimization Problems. *Computers in Industry*, 41(2):113–127, 2000.

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

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

Deb K. and Agrawal R.B. Simulated binary crossover for continuous search space. *Complex Systems*, 9:115–148, 1995.

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

Kannan B. and Kramer S. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design*, (116):405–411, 1995.

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

Liepins G.E. and Potter W.D. 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, New York, 1991.

M.-Montes E., Coello C., and Ricardo L. Engineering optimization using a simple evolutionary algorithm. In *Proceedings of the 15th International Conference on Tools with Artificial Intelligence, ICTA 2003*, pages 149–156. CA, USA, 2003.

Michalewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, New York, 1992.

M.M. Efrén C.C. and Ricardo L. Engineering optimzation using a simple evolutionary algorithm. In *15th International Conference on Tools with Atificail Intelligence – ICTAI'2003*, pages 149–156. California, USA, 2003.

Muhlenbein H., Schomisch M., and Born J. The parallel genetic algorithm as function optimizer. *Parallel Computing*, 17(6-7):619–632, 1991.

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

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

Sandgren E. Nonlinear integer and discrete programming in mechanical design. In *Proc. of the ASME Design Technology Conference*, pages 95–105. Kissimee, FL, 1988.

Shoenauer M. and Michalewicz Z. 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. Springer-Verlag, Berlin, 1996. LNCS.

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

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