Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Information Sciences 156 (2003) 215–251

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

A new adaptive penalty scheme for genetic algorithms

Helio J.C. Barbosa [a,*], Afonso C.C. Lemonge [b]

[a] LNCC/MCT, Rua Getúlio Vargas 333, 25651 070 Petrópolis RJ, Brazil
[b] Departamento de Estruturas, Faculdade de Engenharia, Universidade Federal de Juiz de Fora, Campus Universitário, Bairro Martelos, 36036 330 Juiz de Fora, MG, Brazil

Received 15 March 2002; accepted 1 January 2003

Abstract

A parameter-less adaptive penalty scheme for genetic algorithms applied to constrained optimization problems is proposed. The performance of this new scheme is examined using test problems from the evolutionary computation literature as well as structural engineering constrained optimization problems.

1. Introduction

Evolutionary algorithms (EAs) are weak search algorithms which can be directly applied to unconstrained optimization problems, $\mathrm{UOP}(f, S)$, where one seeks for an element x belonging to the search space S, which minimizes (or maximizes) the real function f. In this case, the EA usually employs a fitness function closely related to f. Two classes of constrained optimization problems are: (i) constraint satisfaction problems, $\mathrm{CSP}(S, b)$, where an element $x \in S$ is sought such that the Boolean function $b(x)$ is evaluated as true (all constraints are simultaneously satisfied) and (ii) constrained optimization problems, $\mathrm{COP}(f, S, b)$, where one seeks for $x \in S$ that minimizes (or maximizes) f in S and is such that $b(x)$ is true. Elements in S satisfying $b(x)$ are called feasible.

* Corresponding author.
E-mail addresses: hcbm@lncc.br (H.J.C. Barbosa), lemonge@numec.ufjf.br (A.C.C. Lemonge).

0020-0255/$ - see front matter
doi:10.1016/S0020-0255(03)00177-4

In this paper we are mainly concerned with COPs and, without loss of generality, only minimization problems will be considered.

The straightforward application of EAs to COPs is not possible due to the additional requirement that a set of constraints must be satisfied. Several difficulties may arise:
- the objective function may be undefined for some or all infeasible elements,
- the check for feasibility can be more expensive than the computation of the objective function value,
- 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. One can have $f(x_1) > f(x_2)$ and $v(x_1) = v(x_2)$ or $f(x_1) = f(x_2)$ and $v(x_1) > v(x_2)$ and still have x_1 closer to the optimum.

It is important to note that most comparisons in the literature have been conducted in problems with constraints which can be written as $g_i(x) \leqslant 0$, where each $g_i(x)$ is a given *explicit* function of the independent (design) variable $x \in R^n$. However, that has to do more with easier reproducibility of results than with practical significance. Although the available test problems attempt to represent different types of difficulties one is expected to encounter when dealing with practical situations, very often the constraints cannot be put explicitly in the form $g_i(x) \leqslant 0$. In engineering design problems most constraints are only known as complex implicit functions of the design variables. In order to check if a certain constraint (the stress level at a particular component, for example) has been violated one is required to perform a whole computational simulation which is carried out by a specific code (usually much more complicated than the GA code) and expending considerable computational resources.

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

Direct techniques comprise (a) the design of special *closed* genetic operators, (b) the use of special decoders, (c) repair techniques and (d) "death penalty".

In special situations, closed genetic operators (in the sense that when applied to feasible parents they produce feasible offspring) can be designed if enough domain knowledge is available [1]. Special decoders [2]—that always generate feasible individuals from any given genotype—have been devised but no applications considering implicit constraints have been published.

Repair methods [3,4] use domain knowledge in order to move an infeasible offspring into the feasible set. However there are situations when it is very expensive, or even impossible, to construct such a repair operator, drastically reducing the range of applicability of repair methods. The design of efficient repair methods constitutes a formidable challenge, specially when implicit constraints are present.

Discarding any infeasible element generated during the search process (“death penalty”) is common practice in non-populational optimization methods. Although problem independent, no consideration is made for the potential information content of any infeasible individual.

Summarizing, direct techniques are problem dependent (with the exception of the “death penalty”) and actually of extremely reduced practical applicability.

Indirect techniques comprise (a) the use of Lagrange multipliers as in [5], which may also lead to the introduction of a “population of multipliers” and to the use of the concept of coevolution as in [6], (b) the use of fitness as well as constraint violation values in a multi-objective optimization setting [7], (c) the use of special selection techniques [8], and (d) “lethalization”: any infeasible offspring is just assigned a given very low fitness value [9].

For other methods proposed in the evolutionary computation literature see [1,10–13] and references therein.

Methods to tackle COPs which require the knowledge of constraints in explicit form have thus limited practical applicability. This fact, together with simplicity of implementation are perhaps the main reasons why penalty techniques, in spite of their shortcomings, are the most popular ones.

Penalty techniques, originating in the mathematical programming community, range from simple schemes (like “lethalization”) to penalty schemes involving from one to several parameters. Those parameters can remain constant (the most common case) or be dynamically varied along the evolutionary process according to an exogenous schedule or an adaptive procedure. Penalty methods, although quite general, require considerable domain knowledge and experimentation in each particular application in order to be effective.

We are interested in developing a general penalty method which

- handles inequality as well as equality constraints,
- does not require the knowledge of the explicit form of the constraints as a function of the decision/design variables,
- is free of parameters to be set by the user,
- can be easily implemented within an existing GA code and
- is robust.

Also, further efficiency can be obtained by adding domain knowledge in the form of

- a better coding,
- special operators,
- special decoders and
- repair operators,

since all results presented here were obtained with

- binary coding,
- standard crossover and mutation operators,
- no GA parameter tuning and
- no use of the explicit form of the constraints in the construction of operators to use such knowledge.

In this paper, in contrast with previous approaches where a single penalty parameter is used for all constraints in a given problem, an adaptive scheme is proposed which automatically sizes the penalty parameter corresponding to *each* constraint along the evolutionary process.

In Section 2, the penalty method and some of its implementations within genetic algorithms are presented. In Section 3 the proposed adaptive scheme is discussed, Section 4 presents numerical experiments with several test-problems from the literature and the paper closes with some conclusions.

2. The penalty method

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 $g_p(x) \geqslant 0$, $p = 1, 2, \ldots, \bar{p}$ as well as equality constraints $h_q(x) = 0$, $q = 1, 2, \ldots, \bar{q}$. Additionally, the variables may be subject to bounds $x_i^L \leqslant x_i \leqslant x_i^U$ but this type of constraint is trivially enforced in a GA and need not be considered here.

Penalty techniques can be classified as *multiplicative* or *additive*. In the multiplicative case, a positive penalty factor $p(v(x), T)$ is introduced in order to amplify the value of the fitness function of an infeasible individual in a minimization problem. One would have $p(v(x), T) = 1$ for a feasible candidate solution x and $p(v(x), T) > 1$ otherwise. Also, $p(v(x), T)$ increases with the "temperature" T and with constraint violation. An initial value for the temperature is required as well as the definition of a function such that T grows with the generation number. This type of penalty has received much less attention in the EC community than the additive type. In the additive case, a penalty functional is added to the objective function in order to define the fitness value of an infeasible element. They can be further divided into (a) *interior* techniques, when a barrier functional $B(x)$—which grows rapidly as x approaches the boundary of the feasible domain—is added to the objective function

$$F(x) = f(x) + \frac{1}{k} B(x)$$

and (b) *exterior* techniques, where a penalty functional is introduced

$$F(x) = f(x) + kP(x) \tag{1}$$

such that $P(x) = 0$ if x is feasible and $P(x) > 0$ otherwise (for minimization problems). In both cases (a) and (b), as $k \to \infty$, the sequence of minimizers of the UOP(F, S) converges to the solution of the COP(f, S, b).

At this point it is useful to define the amount of violation of the jth constraint by the candidate solution $x \in R^n$ as

$$v_j(x) = \begin{cases} |h_j(x)| & \text{for an equality constraint} \\ \max\{0, -g_j(x)\} & \text{otherwise} \end{cases}$$

It is also common to design penalty functions that grow with the vector of violations $v(x) \in R^m$ where $m = \bar{p} + \bar{q}$ is the number of constraints to be penalized. The most popular penalty function is given by

$$P(x) = k \sum_{j=1}^{m} (v_j(x))^{\beta} \tag{2}$$

where k is the penalty parameter and $\beta = 2$. Although it is easy to obtain the unconstrained problem, the definition of a good penalty parameter k is usually a very time-consuming trial-and-error process.

Among the many suggestions in the literature [14,10,15] some of them—more closely related to the work presented here—will be briefly discussed in the following.

2.1. Some methods in the literature

Besides the widely used case of a single constant penalty parameter k, several other proposals can be found in the literature.

2.1.1. Two-level penalties

Le Riche et al. [16] present a GA where two fixed penalty parameters k_1 and k_2 are used independently in two different populations. The idea is to create two sets of candidate solutions where one of them is evaluated with the parameter k_1 and the other with the parameter k_2. With $k_1 \gg k_2$ there are two different levels of penalization and there is a higher chance of maintaining feasible as well as infeasible individuals in the population and to get offspring near the boundary between the feasible and infeasible regions. The strategy can be summarized as (i) create $2 \times$ pop individuals randomly ('pop' is the population size); (ii) evaluate each individual considering each penalty parameter and create two ranks; (iii) combine the two ranks in a single one with size pop; (iv) apply genetic operators; (v) evaluate new offspring ($2 \times$ pop), using both penalty parameters and repeat the process.

2.1.2. Multiple coefficients

Homaifar et al. [17] proposed different penalty coefficients for different levels of violation of each constraint. The fitness function is written as

$$F(x) = f(x) + \sum_{j=1}^{m} k_{ij}(v_j(x))^2$$

where i denotes one of the l levels of violation defined for the jth constraint. This is an attractive strategy because, at least in principle, it allows for a good control of the penalization process. The weakness of this method is the large number, $m(2l+1)$, of parameters that must be set by the user for each problem.

2.1.3. Dynamic coefficients

Joines and Houck [18] proposed that the penalty parameters should vary dynamically along the search according to an exogenous schedule. The fitness function $F(x)$ was written as in (1) and (2) with the penalty parameter, given by $k = (C \times t)^{\alpha}$, increasing with the generation number t. They used both $\beta = 1$ and 2 and suggested the values $C = 0.5$ and $\alpha = 2$.

2.1.4. Adaptive penalties

A procedure where the penalty parameters change according to information gathered during the evolution process was proposed by Bean and Alouane [19]. The fitness function is again given by (1) and (2) but with the penalty parameter $k = \lambda(t)$ adapted at each generation by the following rules:

$$\lambda(t+1) = \begin{cases} \left(\dfrac{1}{\beta_1}\right)\lambda(t) & \text{if } b^i \in \mathscr{F} \text{ for all } t-g+1 \leqslant i \leqslant t \\ \beta_2\lambda(t) & \text{if } b^i \notin \mathscr{F} \text{ for all } t-g+1 \leqslant i \leqslant t \\ \lambda(t) & \text{otherwise} \end{cases}$$

where b^i is the best element at generation i, $\mathscr{F}$ is the feasible region, $\beta_1 \neq \beta_2$ and $\beta_1, \beta_2 > 1$. In this method the penalty parameter of the next generation $\lambda(t+1)$ decreases when all best elements in the last g generations were feasible, increases if all best elements were infeasible and otherwise remains without change.

The method proposed Coit et al. [20], uses the fitness function:

$$F(x) = f(x) + (F_{\text{feas}}(t) - F_{\text{all}}(t)) \sum_{j=1}^{m} (v_j(x)/v_j(t))^{\alpha}$$

where $F_{\text{all}}(t)$ corresponds to the best solution, until the generation t (without penalty), F_{feas} corresponds to the best feasible solution and α is a constant.

Hamida and Schoenauer [21] proposed an adaptive scheme using (i) a function of the proportion of feasible individuals in the population, (ii) a seduction/selection strategy to mate feasible and infeasible individuals and (iii) a selection scheme to give advantage for a given number of feasible individuals.

Runarsson and Yao [8] presented a novel approach where a good balance between the objective and the penalty function values is sought stochastically by a stochastic ranking. Using real-coding in an evolution strategy setting very good results were obtained.

Recently, Wright and Farmani [22] proposed a method that requires no parameters and represents the constraint violation by a single infeasibility measure.

3. The proposed method

In this work a method without any type of user defined penalty parameter is proposed. An adaptive scheme is developed that uses information from the population, such as the average of the objective function and the level of violation of each constraint during the evolution, in order to define different penalties for different constraints. 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. One indication of such difficulty is the number of elements violating a given constraint and the amount of violation. In order to achieve the desired distribution, the jth coefficient is made proportional to the average of the violation of the jth constraint by the elements of the current population.

The fitness function proposed is written as

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

The penalty parameter is defined at each generation by

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

where $\langle f(x)\rangle$ is the average of the objective function values in the current population and $\langle v_l(x)\rangle$ is the violation of the lth constraint averaged over the current population. Denoting by 'pop' the population size, one could also write

$$k_j = \frac{\left|\sum_{i=1}^{\text{pop}} f(x^i)\right|}{\sum_{l=1}^{m}\left[\sum_{i=1}^{\text{pop}} v_l(x^i)\right]^2} \sum_{i=1}^{\text{pop}} v_j(x^i) \tag{4}$$

With the proposed definition one can prove the following properties.

Property 1. *An individual whose jth violation equals the average of the jth violation in the current population for all j, has a penalty equal to the absolute value of the average fitness function of the population.*

Proof. In fact, let $\tilde{x}$ be such an element. By definition,

$$F(\tilde{x}) = f(\tilde{x}) + \sum_{j=1}^{m} \frac{|\langle f(x)\rangle|\langle v_j(x)\rangle}{\sum_{l=1}^{m}[\langle v_l(x)\rangle]^2} v_j(\tilde{x})$$

But, by hypothesis, $v_j(\tilde{x}) = \langle v_j(x)\rangle$ for all j and the result follows:

$$F(\tilde{x}) = f(\tilde{x}) + \frac{|\langle f(x)\rangle|}{\sum_{l=1}^{m}[\langle v_l(x)\rangle]^2} \sum_{j=1}^{m}[\langle v_j(x)\rangle]^2 = f(\tilde{x}) + |\langle f(x)\rangle| \qquad \square$$

Property 2. *The average of the fitness function equals* $\langle f(x)\rangle + |\langle f(x)\rangle|$.

Proof. Computing the average fitness one has

$$\begin{aligned}\langle F(x)\rangle &= \langle f(x)\rangle + \frac{1}{\text{pop}} \sum_{i=1}^{\text{pop}} \sum_{j=1}^{m} \frac{|\langle f(x)\rangle|}{\sum_{l=1}^{m}[\langle v_l(x)\rangle]^2} \langle v_j(x)\rangle v_j(x^i) \\ &= \langle f(x)\rangle + \frac{|\langle f(x)\rangle|}{\sum_{l=1}^{m}[\langle v_l(x)\rangle]^2} \sum_{j=1}^{m}[\langle v_j(x)\rangle]^2 = \langle f(x)\rangle + |\langle f(x)\rangle| \qquad \square\end{aligned}$$

In the next section several examples from the literature are considered in order to test the robustness of the proposed parameter-less scheme. It should be emphasized that the accuracy of the final results of the search depends also on other components of the algorithm not considered here (such as coding, operators, selection scheme, etc.) besides the penalization procedure itself.

4. Numerical experiments

In order to investigate the robustness of the proposed penalty procedure, several optimization problems from the literature, listed in Table 29, are solved using a simple generational GA with Gray code, rank-based selection and elitism (the best element is always copied into the next generation along with nine copies where one randomly chosen bit has been changed). A total of 20 independent runs were executed for each test-problem.

No parameter tuning was attempted and a population size of 200 was used in all cases except in the engineering applications. The operation of recombination was applied with probability $p_r = 0.8$. Standard one-point, two-point and

Table 1
Labels used for each method from the literature

Label	Coding	Reference
KD	Real	[24]
GENECOP I	Binary	[23]
GENECOP II	Binary	[23]
J&H	Real	[13]
HB	Binary	[6]
W&F	Binary	[22]
K&M	Binary	[2]
H&S	Real	[21]
R&Y	Real	[8]

uniform crossover operators were applied each one with its respective relative probability (in this work, $p_c^1 = 0.2$, $p_c^2 = 0.4$ and $p_c^u = 0.4$). Mutation was applied bit wise to the offspring with rate $p_m = 0.03$. All real variables were coded with 50 bits except in problems 7 and 8 and in the engineering applications.

So as to investigate the performance of the adaptive penalty parameters, additional experiments are performed. In these tests a static penalty method is used where a constant parameter is adopted for each constraint. The values of each constant parameter were taken as the average of each k_j, in the best run (i.e. the one which provided the best result), of the adaptive method proposed.

For each test-problem, a comparison is made among the results obtained by (i) the proposed adaptive penalty method (APM), (ii) the same GA using constant penalty parameters (CPM) and (iii) other results from the literature. Table 1 displays the abbreviations used when making references to results from the literature as well as the type of coding adopted.

In the following tables 'NFE' means the number of function evaluations performed in each run of a given method.

4.1. Test-problem 1

The first example to be investigated, from [24], is a two dimensional minimization problem subject to two nonlinear inequality constraints. The search space is bounded by $0 \leqslant x_i \leqslant 6$, $i = 1, 2$ and the optimum solution is $(x_1, x_2; f^*) = (2.246826, 2.381865; 13.59085)$. The number of generations allowed was 800 and the best solution found was (2.246819504, 2.381745171; 13.5908412) the worst one was (2.246702559, 2.379592430; 13.5910134) and the average value was $f(x) = 13.5908789$. The average of the penalty coefficients in the best run were: $k_1 = 673.38$ and $k_2 = 353.00$ (Fig. 1). Using these values as constant parameters for the CPM procedure, the best solution achieved was (2.246826120, 2.381868195; 13.5908408), the worst one was (2.246913738, 2.383508992; 13.5909314) and the average value was $f(x) = 13.5908707$. Table 2

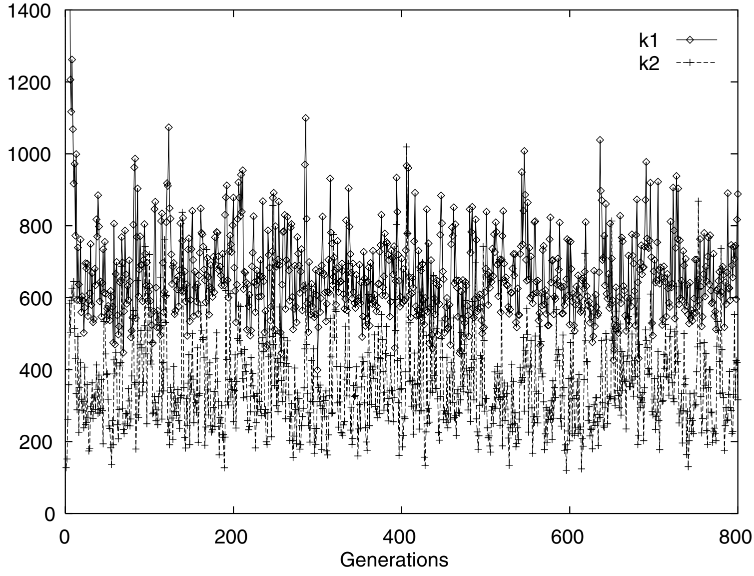

Fig. 1. Penalty parameters—test-problem 1.

Table 2
Comparison of results on TP1 (optimum is 13.59085)

	APM	CPM	KD	KD*	APM*	CPM*
Best	13.5908412	13.5908408	13.59085	13.59108	13.5909591	13.5908635
Average	13.5908789	13.5908707	13.61673	16.35284	13.5977691	15.1855761
Worst	13.5910134	13.5909314	117.02971	172.81369	13.6248564	45.3294040
NFE (10^3)	160	160	2.5	2.5	2.5	2.5

presents a comparison of results for this test-problem where KD* denotes the results achieved when the Powell and Skolnick's [25] method was implemented by Deb using a penalty parameter $R = 1$. APM* and CPM* denote the results using the same number of evaluations performed in KD and show the same level of solution accuracy and computational efficiency like those achieved in KD. The values of variables x_1 and x_2 are, respectively, (2.246724017, 2.379984569) (best with APM*), (2.244876511, 2.351811002) (worst with APM*), (2.246866889, 2.382630658) (best with CPM*) and, finally, (1.911155340, 1.394320470) (worst with CPM*).

4.2. Test-problem 2

The function considered here, taken from Michalewicz [23], is to be minimized over the set $[2, 50] \times [0, 50]$. The known global solution is (15.811388,

Table 3
Comparison of results on TP2 (optimum is 5)

	APM	CPM	GENECOP I	J&H
Best	5.0000002	5.0000001	4.999595375	5.00000
Average	5.0001146	5.0001018	–	5.00002
Worst	5.0004391	5.0007294	–	5.00005
NFE (10^4)	16	16	–	1.6003

1.581139; 5) The best feasible solution was (15.813464100, 1.580931293; 5.0000002) the worst one was (15.916499553, 1.570697176; 5.0004391) and the average was $f(x^*) = 5.0001146$. The average values of the k_j were $k_1 = 3254.74$ and $k_2 = 152.02$. Using the CPM scheme the best solution achieved was (15.812927658, 1.580984938; 5.0000001), the worst one was (15.676930358, 1.594699967; 5.0007294) and the average value was $f(x) = 5.0001018$. Table 3 presents a comparison of results for this test-problem.

Fig. 2 displays the evolution of the penalty coefficients (in the best run) where it is clear that the method assigns a much higher value to k_1 which is the only constraint active at the optimum.

4.3. Test-problem 3

This test-problem, from Michalewicz [23], is also a minimization problem (see Table 29). The global optimum is (1, 1; 1) and it was found in all runs. The average values of the penalty coefficients were (in the best run) $k_1 = 39.59$ and $k_2 = 9.59$ (Fig. 3). Using the CPM scheme all runs were also able to find the optimum solution. Table 4 presents a comparison of results for this test-problem.

Again, distinct level of penalty coefficients were observed.

4.4. Test-problem 4

This test, taken from Michalewicz [23], corresponds to the minimization of $f(x)$ (see Table 1) over the set $[-0.5, 0.5] \times (-\infty, 1]$. The optimal solution is (0.5, 0.25; 0.25) The number of generations allowed was 200. The best, worst and average values found were $f(x) = 0.25$ corresponding to the global optimum. The average values of the penalty coefficients were $k_1 = 2007.56$ and $k_2 = 949.82$ (Fig. 4). Using the CPM scheme, all runs were also able to find the optimal solution. Table 5 presents a comparison of results for this test-problem.

Once more, two distinct levels of penalization are observed.

4.5. Test-problem 5

In this test-problem, from Michalewicz [23], the function is to be minimized over the set $S = [0, 3] \times [0, 4] \subset R^2$. The global solution is (2.329520197,

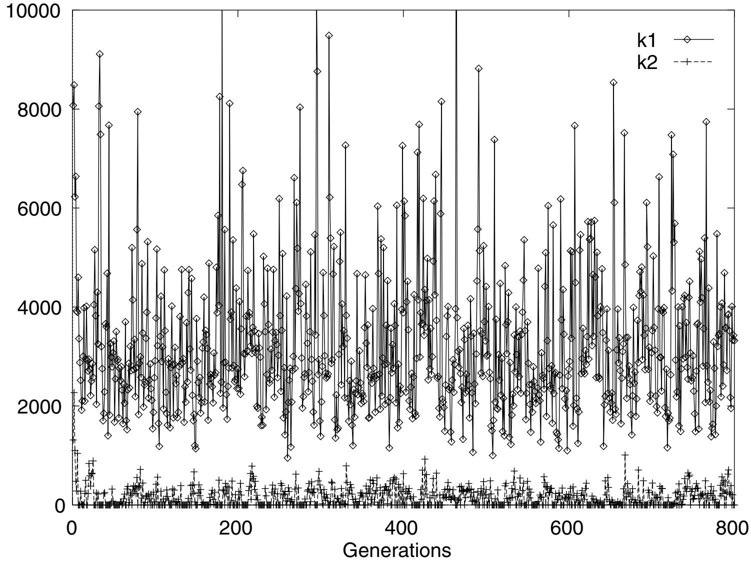

Fig. 2. Penalty parameters—test-problem 2.

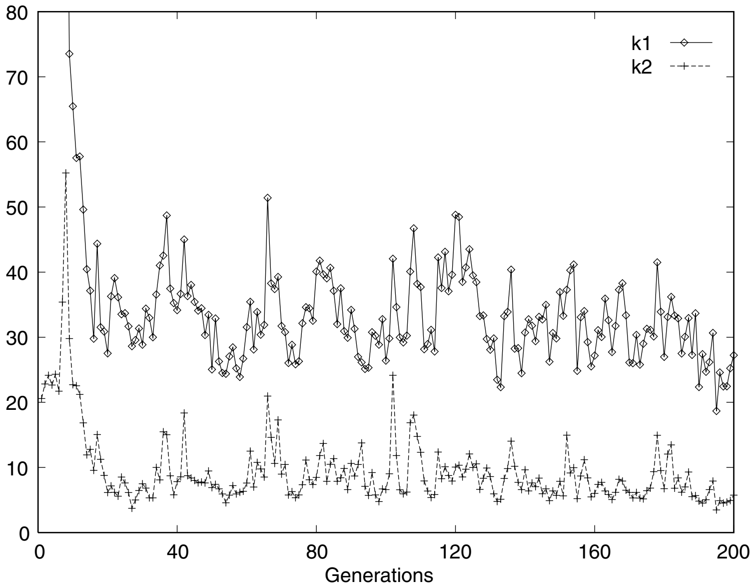

Fig. 3. Penalty parameters—test-problem 3.

Table 4
Comparison of results on TP3 (optimum is 1)

	APM	CPM	GENECOP II	J&H
Best	1.0	1.0	0.999942002	1.0
Average	1.0	1.0	–	1.0
Worst	1.0	1.0	–	1.0
NFE (10^3)	4	4	–	3.73

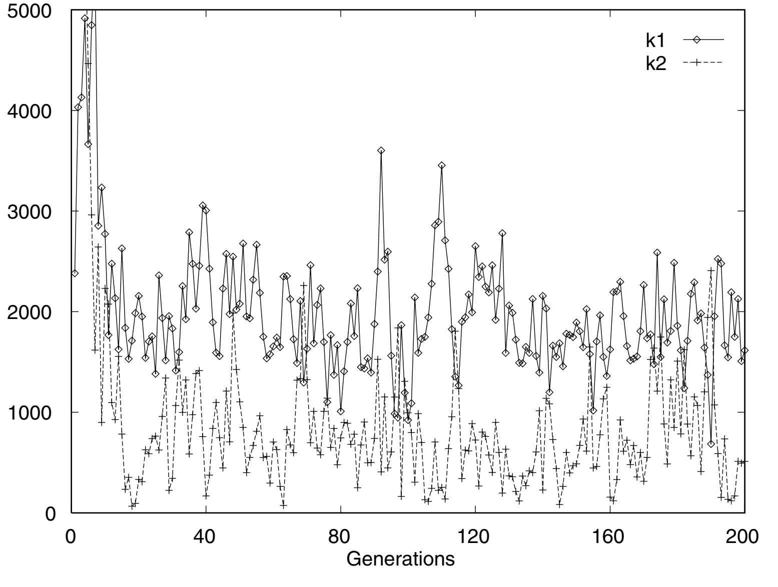

Fig. 4. Penalty parameters—test-problem 4.

Table 5
Comparison of results on TP4 (optimum is 0.25)

	APM	CPM	GENECOP II	J&H
Best	0.25	0.25	0.25	0.25
Average	0.25	0.25	–	0.25
Worst	0.25	0.25	–	0.25
NFE (10^3)	4	4	–	4.658

3.178493074; −5.508013271) and both constraints are active. The number of generations allowed was 1000. The best feasible solution found was (2.329520235, 3.178492879; −5.50801311), the worst feasible solution was (2.329525600, 3.178467607; −5.50799321) and the average value was −5.50801157. The average value of the penalty coefficients were: $k_1 = 182.58$

and $k_2 = 455.00$. Using the CPM scheme the best solution achieved was (2.329520235, 3.178492879*i*; −5.50801311), the worst one was (2.329525600, 3.178467607; −5.50799321) and the average was $f(x) = -5.50801157$. Table 6 presents a comparison of results for this test-problem.

Again, two well-defined levels of penalization emerge in Fig. 5.

4.6. Test-problem 6

In this test problem, from Michalewicz [23], the function $f(x, \mathbf{y})$ is to be minimized under five linear constraints over the set $y_3 \leqslant 1$, $y_4 \leqslant 1$, $y_5 \leqslant 2$, $x \geqslant 0$ and $y_i \geqslant 0$ for $1 \leqslant i \leqslant 5$. The global solution is $(x, \mathbf{y}^*) = (0, 6, 0, 1, 1, 0)$ with $f(x, \mathbf{y}^*) = -11$. The number of generations allowed was 2000. The best

Table 6
Comparison of results on TP5 (optimum is −5.50801327)

	APM	CPM	HB	GENECOP II	J&H
Best	−5.50801311	−5.50801311	−5.50801311	−5.5085	−5.50801
Average	−5.50801157	−5.50801157	−5.50790070	–	−5.50801
Worst	−5.50799321	−5.50799321	−5.50578524	–	−5.50801
NFE (10^3)	200	200	135	–	3.72

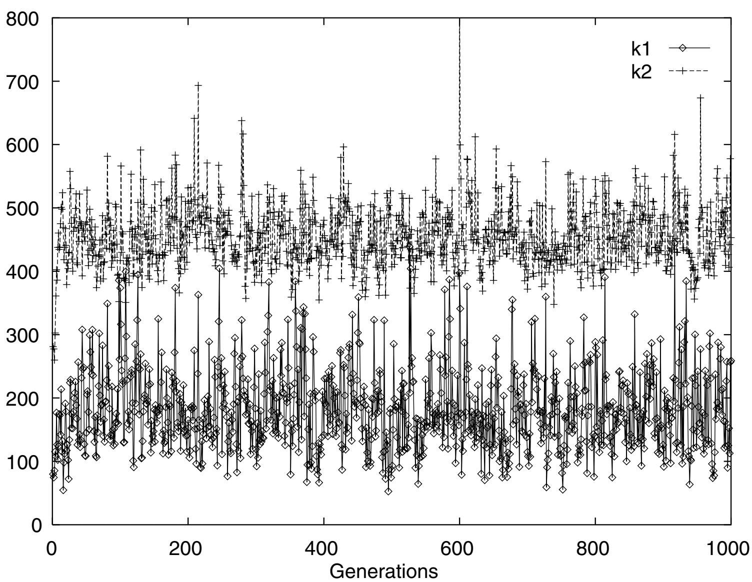

Fig. 5. Penalty parameters—test-problem 5.

Table 7
Comparison of results on TP6 (optimum is −11.0)

	APM	CPM	GENECOP I
Best	−10.9971371	−10.9959432	−11.0
Average	−10.9779104	−10.9734209	−11.0
Worst	−10.9804996	−10.9334936	−11.0
NFE (10^4)	40	40	7

feasible solution found was $(x, \mathbf{y}^*) = (0.000008047, 5.998867929, 0.000724494, 0.999996722, 0.998281717, 0.000318885)$ corresponding to $f(x, \mathbf{y}^*) = -10.9971371$, the worst feasible solution was $(x, \mathbf{y}^*) = (0.000000000, 5.962490021, 0.007517040, 1.000000000, 0.999999106, 0.002977252)$ corresponding to $f(x, \mathbf{y}^*) = -10.9804996$ and the average corresponding to $f(x, \mathbf{y}^*) = -10.9779104$. The average values of k_j values are $k_1 = 23.93$, $k_2 = 0.0447$, $k_3 = 0.0001$ and $k_4 = 1.7576$. Using the CPM scheme the best solution achieved was $(x, \mathbf{y}^*) = (0.000000000, 5.993473708, 0.001381636, 0.999987125, 0.999530256, 0.000684321)$ with $f(x, \mathbf{y}^*) = -10.9959432$, the worst was $(x, \mathbf{y}^*) = (0.000000000, 5.887711224, 0.002608597, 0.999999911, 0.999874085, 0.040817322)$ with $f(x, \mathbf{y}^*) = -10.9334936$ and the average is $f(x, \mathbf{y}^*) = -10.9734209$. Table 7 presents a comparison of results for this test-problem.

It can be seen that the highest level of penalization corresponds to the first constraint—which is the one active at the optimum—and the lowest level is that of the more easily satisfied one, namely, constraint number three.

4.7. Test-problem 7

This test corresponds to the function $G_1(x)$ in Koziel and Michalewicz [2] which is to be minimized under nine linear constraints (see Table 29) and bounds $x_i \in [0, 1]$, $i = 1, \ldots, 9$, $x_i \in [0, 100]$, $i = 10, 11, 12$ and $x_{13} \in [0, 1]$. The global solution is (1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1; −15). Six out of the nine constraints are active at the global optimum. The number of bits per variable was 25, except for the variables x_1 to x_9 and x_{13} which were coded with 10 bits and for the variables x_{10} to x_{12} coded with 20. The number of generations allowed was 800.

All runs produced the optimal solution. The average values of the penalty coefficients were: $k_1 = 20.93$, $k_2 = 21.01$, $k_3 = 20.80$, $k_4 = 11.75$, $k_5 = 11.49$, $k_6 = 11.56$, $k_7 = 10.34$, $k_8 = 10.02$ and $k_9 = 10.10$. Using the CPM scheme all runs were also able to find the optimal solution. Table 8 presents a comparison of results for this test-problem.

Figs. 6 and 7 display the evolution of the penalty coefficients in the best run, showing that the first three constraints are more heavily penalized.

Table 8
Comparison of results on TP7 (optimum is −15)

	APM	CPM	W&F	HB	K&M	KD	H&S	R&Y
Best	−15.0	−15.0	−14.9996	−15.0	−14.7864	−15.0	−15.0	−15.0
Average	−15.0	−15.0	−14.988	−14.911145	−14.7082	−15.0	−15.0	−15.0
Worst	−15.0	−15.0	−14.9081	−14.227718	−14.6154	−13.0	–	−15.0
NFE (10^4)	16	16	140	16	140	6.5	20	3.5

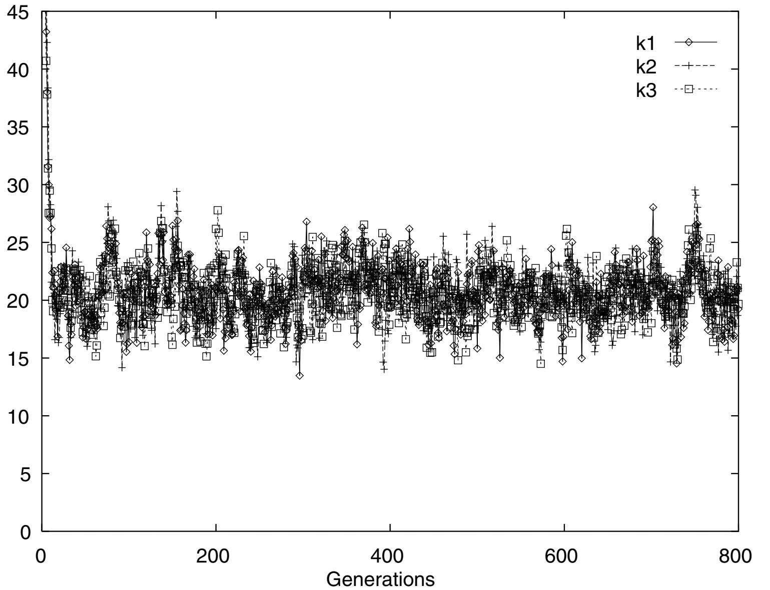

Fig. 6. Penalty parameters—test-problem 7.

4.8. Test-problem 8

This test corresponds to the function $G_3(x)$ in Koziel and Michalewicz [2] which is to be maximized under 1 nonlinear equality constraint (see Table 29) and bounds $x_i \in [0, 1]$, $i = 1, \ldots, n$, where $n = 10$. The global maximum is reached for $x_i^* = 1/\sqrt{n}$, $i = 1, \ldots, n$ leading to $G_3(x^*) = 1$. The equality constraint was converted into one inequality constraint bounding the absolute value of the degree of violation by 0.001 (i.e., $|h(x)| \leqslant 0.001$). The number of bits per variable was 10.

The best feasible solution found was $G_3(x^*) = 1.0010096$ (which is actually slightly infeasible), the worst solution was $G_3(x^*) = 0.9805778$ (feasible) and the average $G_3(x^*) = 0.9937453$.

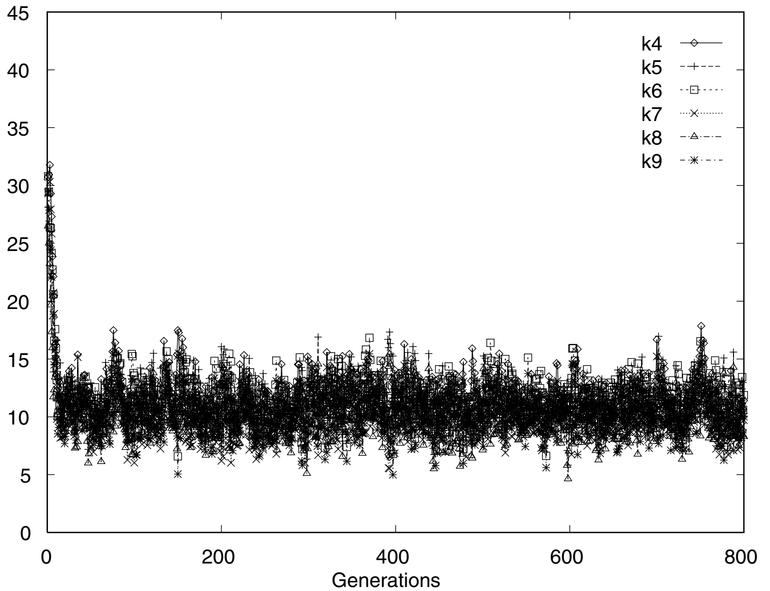

Fig. 7. Penalty parameters—test-problem 7.

The evolution of the penalty coefficient in the best run is displayed in Fig. 8. The average value of k_1 in this run was 433.38. However, using the CPM scheme with this *fixed* average value for k_1, no feasible solutions were found, indicating the importance of the adaptive process within APM.

Table 9 presents a comparison of results for this test-problem.

The evolution of the penalty coefficient in the best run is displayed in Fig. 8.

4.9. Test-problem 9

This test corresponds to the function $G_4(x)$ in Koziel and Michalewicz [2] which is to be minimized under three nonlinear inequality constraints (see Table 29) and bounds $x_1 \in [72, 102]$, $x_2 \in [33, 45]$ and $x_i \in [27, 45]$ for $i = 3, 4, 5$. Two constraints (upper bound of the first inequality and the lower bound of the third inequality) are active at the optimum. Each inequality constraint was converted into two inequality constraints resulting in a total of six constraints. The global minimum is (78, 33, 29.995256025682, 45, 36.775812905788; −30 665.539). The number of bits per variable was 25.

The best feasible solution found was (78.00, 33.00, 29.9960221, 44.9994437, 36.7741062; −30 665.4033720), the worst solution was (78.00, 33.1092614, 30.0579779, 44.9999962, 36.6130036; −30 655.3889603) and the average $G_4(x^*) = -30\,662.38818$. The average values of the penalty coefficients were

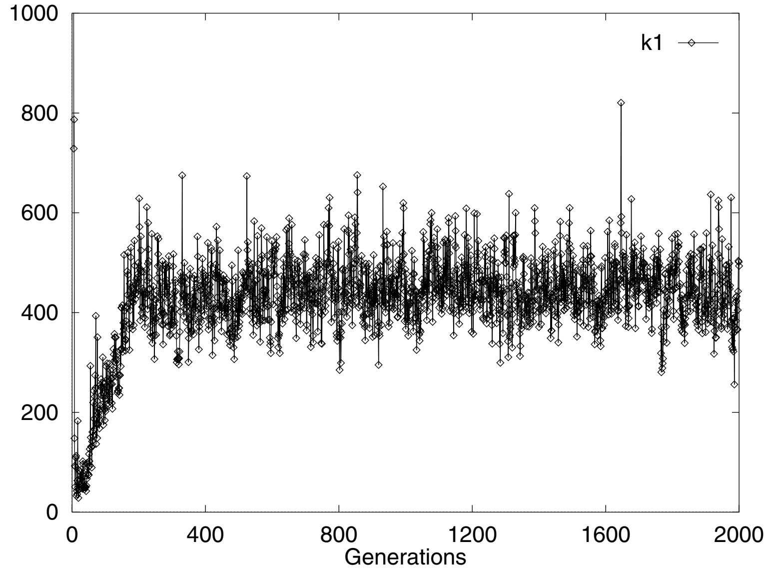

Fig. 8. Penalty parameters—test-problem 8.

Table 9
Comparison of results on TP8 (optimum is 1)

	APM	CPM	W&F	K&M	H&S	R&Y
Best	1.0010096	–	0.9994	0.9997	1.000000	1.000
Average	0.9937453	–	0.99902	0.9989	0.999979	1.000
Worst	0.9805778	–	0.99807	0.9978	–	1.000
NFE (10^5)	4	4	14	14	20	3.5

$k_1 = 3\,348\,195$, $k_2 = 1.0002$, $k_3 = 2187$, $k_4 = 1.0002$, $k_5 = 132\,085$ and $k_6 = 1\,602\,980$. With the CPM scheme the best solution achieved was (78.00, 33.0017681, 29.9968263, 44.9999952, 36.7718462; $-30\,665.2918716$), the worst one was (78.00, 33.1537768, 30.0943058, 44.9997114, 36.5262867; $-30\,649.9138933$) and the average value was $G_4(x^*) = -30\,662.4734500$. Table 10 presents a comparison of results for this test-problem.

4.10. Test-problem 10

This test corresponds to the function $G_5(x)$ in Koziel and Michalewicz [2] which is to be minimized under two linear inequality constraints and three linear equality constraints (see Table 29) and bounds $x_1 \in [0, 1200]$, $x_2 \in [0, 1200]$, $x_3, x_4 \in [-0.55, 0.55]$. The best known global solution is (679.9453,

Table 10
Comparison of results on TP9 (optimum is −30 665.539)

	APM	CPM	W&F	K&M	H&S	R&Y	KD
Best	−30 665.403	−30 665.292	−30 650.01	−30 645.9	−30 665.5	−30 665.539	−30 665.537
Average	−30 662.388	−30 662.473	−30 609.97	−30 655.3	−30 658.9	−30 665.539	−30 665.535
Worst	−30 655.388	−30 649.913	−30 604.2	−30 643.8	–	−30 665.539	−29 846.654
NFE (10^4)	120	120	140	140	200	35	25

1026.067, 0.1188764, −0.3962336; 5126.4981). The number of bits per variable was 25.

The equality constraints were converted into three inequality constraints bounding the absolute value of the degree of violation by 0.001 (i.e., $|h(x)| \leqslant 0.001$).

The best feasible solution found was (688.342365275, 1017.109710488, 0.112895237, −0.399076326; 5126.868508). For the best solution the sum of the all violations is equal to 1.61×10^{-5}, the worst solution was (1096.439263118, 614.950722902, −0.165659941, −0.539923986; 5992.370532) and the average $G_4(x^*) = 5458.360696$. The average values of the penalty coefficients were $k_1 = 0.0591$, $k_2 = 0.8729$, $k_3 = 535.80$, $k_4 = 701.59$ and $k_5 = 501.14$. With the CPM scheme the best solution achieved was (688.525148884, 1016.915160922, 0.112765222, −0.399138252; 5126.885195), the worst one was (1108.777675294, 603.732043616, −0.173828763, −0.544305506; 6043.618826) and the average value was $G_5(x^*) = 5465.868517$. Table 11 presents a comparison of results for this test-problem.

4.11. Test-problem 11

In this test problem, from Koziel and Michalewicz [2], the function $G_8(x)$ is to be maximized over the set $0 \leqslant x_i \leqslant 10$, $x_i = 1, 2$ subject to two nonlinear inequalities. The global optimum solution is located at $x^* = (1.2279713, 4.2453733)$ with $G_8(x^*) = 0.095825$. The number of generations allowed was 2000.

Table 11
Comparison of results on TP10 (optimum is 5126.4981)

	APM	CPM	W&F	K&M	H&S	R&Y
Best	5126.868508	5126.885195	5126.6398	–	4707.52	5126.497
Average	5458.360696	5465.868517	5131.0404	–	4283.41	5128.881
Worst	5992.370532	6043.618826	5135.4409	–	–	5142.472
NFE (10^4)	120	120	140	140	200	35

Table 12
Comparison of results on TP11 (optimum is 0.0958250)

	APM	CPM	W&F	K&M	H&S	R&Y
Best	0.0958250	0.0958250	0.09588	0.0958250	0.0958250	0.0958250
Average	0.0957742	0.0958071	0.09246	0.0891568	0.0958250	0.0958250
Worst	0.0954687	0.0954863	0.06413	0.0291438	–	0.0958250
NFE (10^5)	4	4	14	14	20	3.5

The best feasible solution found was (1.227976804, 4.245381241; 0.0958250), the worst feasible solution was (1.220251148, 4.248304253; 0.0954687) and the average value was 0.0957742. The average values of the penalty coefficients were $k_1 = 114.53$ and $k_2 = 65.44$. With the CPM scheme the best solution achieved was (1.227971440, 4.245373493; 0.0958250), the worst one was (1.225487118, 4.232705717; 0.0954863) and the average value was 0.0958071. Table 12 presents a comparison of results for this test-problem.

4.12. Test-problem 12

This test corresponds to the function $G_9(x)$ in Koziel and Michalewicz [2] to be minimized over the set $[-10, 10]^7$. The optimal solution is (2.330499, 1.951372, −0.4775414, 4.365726, −0.6244870, 1.038131, 1.594227; 680.6300573) with two constraints active. The number of generations allowed was 4000.

The best feasible solution found in 20 runs was (2.299137, 1.950377, −0.4695156, 4.373632, −0.6055483, 1.088105, 1.579723; 680.6668170), the worst feasible solution was (2.367184, 2.048445, −0.1516828, 4.074772, −0.6585822, 1.067886, 1.586243; 682.2892937) and the average value was 681.2387964. The average values of the penalty coefficients were$k_1 = 4586.73$, $k_2 = 310.68$, $k_3 = 212.26$ and $k_4 = 357.40$. Using the CPM scheme the best solution achieved was (2.287484, 1.948796, −0.6203696, 4.393392, −0.6644663, 1.057887, 1.583071; 680.7392373), the worst one was (2.326899, 2.075909, −0.4374930, 3.999864, −0.6199631, 1.105703, 1.580660; 683.0317508) and the average value was 681.4560823. Table 13 presents a comparison of results for this test-problem.

Table 13
Comparison of results on TP12 (optimum is 680.6300573)

	APM	CPM	W&F	HB	K&M	KD	H&S	R&Y
Best	680.6668170	680.7392373	681.1982	680.68	680.91	680.634460	680.630	680.630
Average	681.2387964	681.4560823	684.413	681.33	681.16	680.641724	680.637	680.656
Worst	682.2892937	683.0317508	689.6234	683.18	683.18	680.650879	–	680.763
NFE (10^5)	8	8	14	5.76	14	3.5	20	3.5

4.13. Test-problem 13

This function, taken from Koziel and Michalewicz [12], is to be minimized over the set $S = [-1, 1] \times [-1, 1] \subset R^2$. There are two global solutions (±0.707106781, 0.5; 0.75). The number of generations allowed was 600.

The best feasible solution found in 40 runs was (0.706611505, 0.500299796; 0.7490001), the worst one (−0.702401868, 0.494368359*xi*; 0.7490317) and the average value was 0.7490065.

Table 14 presents a comparison of results for this test-problem. The average value of k_1 in the best run was 168.04. Using the CPM scheme the best solution achieved was (0.706490568, 0.500128910; 0.7490000), the worst one was (−0.701149932, 0.492611214; 0.749054607), and the average value was 0.7490546. Table 14 presents a comparison of results for this test-problem.

Fig. 9 displays the evolution of the penalty parameter.

Table 14
Comparison of results on TP13 (optimum is 0.75)

	APM	CPM	W&F	HB	K&M	H&S	R&Y
Best	0.7490317	0.7490000	0.75	0.74999996	0.75	0.75	0.75
Average	0.7490065	0.7490083	0.808	0.75014149	0.75	0.75	0.75
Worst	0.7490317	0.7490546	0.90296	0.75074318	0.75	–	0.75
NFE (10^5)	1.2	1.2	14	3.6	14	20	3.5

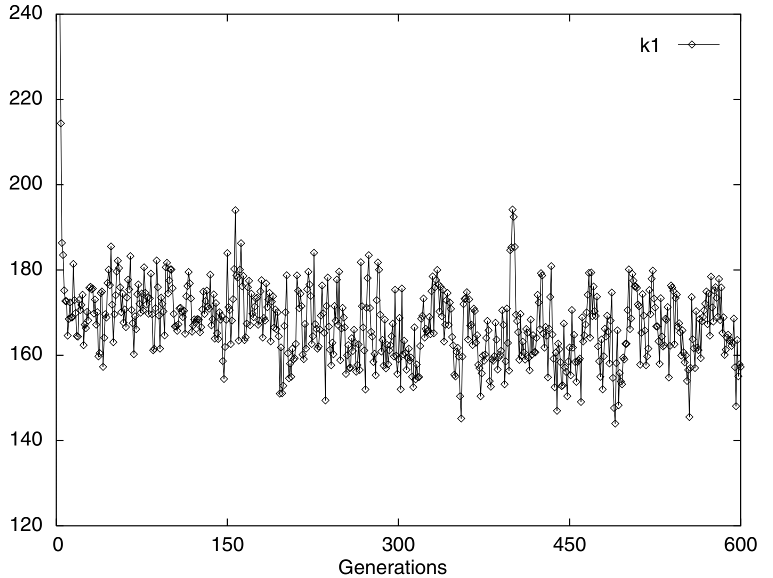

Fig. 9. Penalty parameters—test-problem 13.

4.14. Test-problem 14

All cases studied up to this point involved only continuous variables. This one, from [26], corresponds to a maximization problem involving a nonlinear function and mixed variables. The problem contains three continuous and two discrete variables and three nonlinear inequalities and is written as

$$\begin{aligned} \text{maximize} \quad & f(x_1,x_2,x_3,y_1,y_2) = -5.357854x_1^2 - 0.835689y_1x_3 \\ & \qquad\qquad - 37.29329y_1 + 40\,792.141 \\ \text{subject to} \quad & a_1 + a_2y_2x_3 + a_3y_1x_2 - a_4x_1x_3 \leqslant 92 \\ & a_5 + a_6y_2x_3 + a_7y_1y_2 - a_8x_1^2 - 90 \leqslant 20 \\ & a_9 + a_{10}x_1x_3 + a_{11}y_1x_1 - a_{12}x_1x_2 - 20 \leqslant 5 \end{aligned}$$

The coefficients a_1 to a_{12} are given in Table 15.

The search space is bounded by $27 \leqslant x_1, x_2, x_3 \leqslant 45$ (continuous), $78 \leqslant y_1 \leqslant 102$ and $33 \leqslant y_2 \leqslant 45$ (discrete). The global solution is, for any combination of x_2 and y_2, (27, 27, 78; 32 217.4).

The number of bits per variable was 25, except for the variables y_1 and y_2 which were coded with eight bits each. The number of generations allowed was 150.

All runs produced the optimal solution. The average values of the penalty coefficients were $k_1 = 4\,213\,193$, $k_2 = 253.55$, $k_3 = 124\,353$, With the CPM scheme all runs were also able to find the optimal solution. Table 16 presents a comparison of results for this test-problem.

4.15. Test-problem 15

This function has been considered in test-problem 12 (function $G_9(x)$ in Koziel and Michalewicz [2]), with continuous variables. Here the variables x_1, x_2 and x_3 are considered discrete while the others remain continuous.

Table 15
Coefficients for TP14

a_1	85.3344070
a_2	0.0056858
a_3	0.0006262
a_4	0.0022053
a_5	80.5124900
a_6	0.0071317
a_7	0.0029955
a_8	0.0021813
a_9	9.3009610
a_{10}	0.0047026
a_{11}	0.0012547
a_{12}	0.0019085

Table 16
Comparison of results on TP14 (optimum is 32 217.4)

	APM	CPM
Best	32 217.4275318	32 217.4275318
Average	32 217.4275318	32 217.4275318
Worst	32 217.4275318	32 217.4275318
NFE (10^3)	3	3

For the discrete variables the search space is $x_1 \in \{1, 2, 3, 4, 5\}$, $x_2, x_3 \in \{0, 1, 2, 3, 4, 5\}$ and $x_i \in [0, 10]^7, i = 4, 7$. The number of bits per variable was 25, except for the variables x_1, x_2 and x_3 which were coded with 6 bits The number of generations allowed was 2000.

The best solution was (2, 2, 0, 4.305589625, −0.630482156, 1.132328842, 1.463150724; 682.9761850), the worst solution was (2, 2, 0, 4.297434518, −0.574354845, 1.132328246, 1.463150724; 683.0348242) and the average value was 682.9926149.

The average of values of the penalty coefficients were $k_1 = 60\,977.76$, $k_2 = 17.71$, $k_3 = 44\,199.99$ and $k_4 = 26\,097.52$.

Using the CPM scheme the best solution found was (2, 2, 0, 4.304898212, −0.625719149, 1.132328842, 1.463150724; 682.9760198), the worst solution was (2, 2, 0, 4.293381402, −0.546499209, 1.132328842, 1.463150724; 683.1052835) and the average value was 682.9984019.

Table 17 presents a comparison of results for this test-problem and results from [27] that presented a single result of the function equal to 686.956 using a genetic algorithm. Also, this reference presents others results achieved using others techniques and among them the best result achieved is 686.090.

Table 18 presents a summary of the average values of the penalty coefficients found by the APM in each test-problem.

4.16. Engineering applications

In this section the adaptive penalty scheme proposed is applied to a very important structural engineering optimization problem, namely, that of

Table 17
Comparison of results on TP15

	APM	CPM
Best	682.9761850	682.9760198
Average	682.9926149	682.9984019
Worst	683.0348242	683.1052835
NFE (10^5)	4	4

Table 18
Summary of the average of the penalty coefficients in each test-problem

TP	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9
1	673.38	353.00							
2	3254.74	152.02							
3	39.59	9.59							
4	2007.56	949.82							
5	182.58	455.00							
6	23.93	0.0447	0.0001	1.7576					
7	20.93	21.01	20.80	11.75	11.49	11.56	10.34	10.02	10.10
8	433.38								
9	3 348 195	1.0002	2187	1.0002	132 085	1 602 980			
10	0.0591	0.8729	535.80	701.59	501.14				
11	114.53	65.44							
12	4586.73	310.68	212.26	357.40					
13	168.04								
14	4 213 193	253.55	124 353						
15	60 977.76	17.71	44 199.99	26 097.52					

minimizing the weight of a structure. For framed structures were the bars have a uniform cross-sectional area, the weight is simply written as

$$W = \sum_{k=1}^{m} \rho a_k L_k \tag{5}$$

where L_k is the length of the kth bar of the truss and a_k is the corresponding cross-sectional area. However, in practice, the bars of the structure are grouped by the designer (to maintain symmetry, for instance) in a way that to each group a single cross-sectional area A is assigned (see Tables 20 and 25). The problem is then written as: Find the set of areas $\mathbf{a} = \{A_1, A_2, \ldots, A_l\}$, such that it minimizes the weight of the structure. Although the objective function is very simple (linear with respect to the design variables A_i) the constraints are implicit nonlinear functions of the design variables.

In fact, the minimization is subject to the (normalized) constraints concerning stress:

$$\frac{|\sigma_k|}{\sigma_{max}} - 1 \leqslant 0, \quad k = 1, 2, \ldots, m \tag{6}$$

where m is the number of bars, and the (normalized) constraints concerning all global u_j displacements:

$$\frac{|u_j|}{u_{\max}} - 1 \leqslant 0, \quad j = 1, 2, \ldots, n \tag{7}$$

where n is the number of degrees of freedom of the discrete model.

In order to check if all constraints are satisfied a structural analysis must be performed. Within the linear theory of elasticity and using the finite element method [28] a system of algebraic equilibrium equations is assembled

$$Ku = f \tag{8}$$

where K is the (symmetric, positive-definite) order n stiffness matrix and f is the vector of equivalent nodal forces corresponding to a given load case for the structure. The system is solved for the discretized displacement field u which is then used to check the inequalities (7). Using now both u and the design variables $\mathbf{a}$, one can compute the stress σ_k in each bar and check the inequality constraints (6). A feasible design satisfies all the above constraints while infeasible ones will be penalized.

4.16.1. A ring-shaped truss structure

The first example corresponds to the weight minimization of a 60-bar trussed ring proposed by Patnaik et al. [29] depicted in Fig. 10 (not to scale) under three multiple-load conditions as given in Table 19. The outer radius of

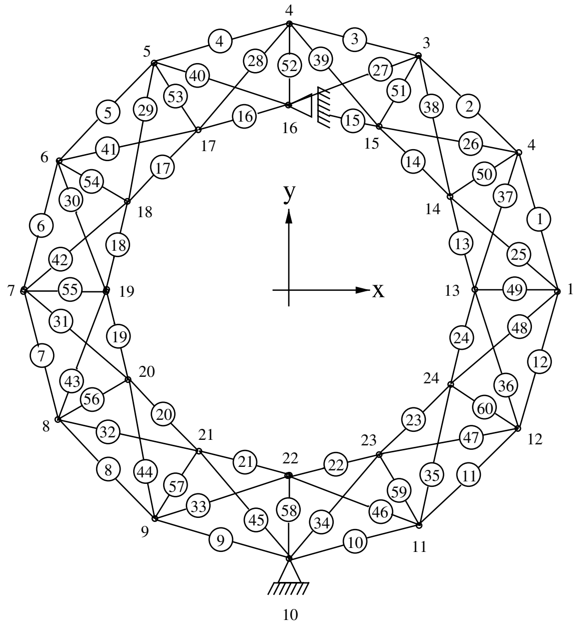

Fig. 10. The 60-bar trussed ring.

Table 19
Load cases for the ring structure (kips)

Load case	Node	F_x	F_y
1	1	−10.0	0
	7	9.0	0
2	15	−8.0	3.0
	18	−8.0	3.0
3	22	−20.0	10.0

the ring is 100 in. and the inner radius is 90 in. and the material has Young's modulus equal to 10^4 ksi and density of 0.1 lb/in.^{3}. There are 198 constraints where 180 refer to allowable stress ($\sigma_i = 60$ ksi, $i = 1$–60), and 18 refer to displacement constraints given as three constraints along of any x and y directions of magnitude: 1.75 in at node 4, 2.25 in at node 13 and 2.75 in at node 19. The cross-sectional areas of the bars are supposed to be continuous with a minimum value equal to 0.5 in.^{2} and they are linked in 25 groups as shown in Table 20. The population size was set equal to 400 and 2000 generations were performed in 20 independent runs and 10 bits were used for each one of the 25 variables.

Table 20
Design variables and corresponding bars for the ring structure

Group	Bars
A_1	49–60
A_2	1, 13
A_3	2, 14
A_4	3, 15
A_5	4, 16
A_6	5, 17
A_7	6, 18
A_8	7, 19
A_9	8, 20
A_{10}	9, 21
A_{11}	10, 22
A_{12}	11, 23
A_{13}	12, 24
A_{14}	25, 37
A_{15}	26, 38
A_{16}	27, 39
A_{17}	28, 40
A_{18}	29, 41
A_{19}	30, 42
A_{20}	31, 41
A_{21}	32, 42
A_{22}	33, 45
A_{23}	34, 46
A_{24}	35, 47
A_{25}	36, 48

Table 21
Comparison of results for the ring structure (the optimum is not known)

	APM	CPM
Best	311.8757	315.4792
Average	333.0190	337.3386
Worst	384.1990	382.9452
NFE (10^4)	80	80

Table 21 shows the results obtained by the APM and CPM schemes and in Table 22 the corresponding design variables are presented. Most results (final weight) from reference [29] are slightly lower than the best produced here. However, in that paper the final values of the design variables are not shown and, as a result, it was not possible to check if all constraints were indeed rigorously satisfied as they were when APM and CPM were used. The value of the design variables found here are listed in Table 23. Table 24 shows the average of the penalty coefficients in the best run.

Table 22
Design variables in the APM and CPM strategies for the ring structure

Var.	APM	CPM
A_1	1.1202346	1.1906158
A_2	2.0219941	2.2771261
A_3	0.5087977	0.6055718
A_4	1.7272727	1.5733138
A_5	1.5205279	1.3753666
A_6	0.5263930	0.5087977
A_7	1.9032258	1.9340176
A_8	2.1275660	2.0527859
A_9	0.9882698	1.2390029
A_{10}	2.0527859	1.8196481
A_{11}	2.0527859	1.6392962
A_{12}	0.7243402	0.5263930
A_{13}	1.9604106	2.1979472
A_{14}	1.2302053	1.2346041
A_{15}	0.9970674	1.0498534
A_{16}	0.6055718	0.7595308
A_{17}	0.7287390	0.6143695
A_{18}	1.0938416	1.1202346
A_{19}	1.1158358	1.1158358
A_{20}	1.1686217	1.1554252
A_{21}	1.0674487	1.1862170
A_{22}	1.0630499	1.0718475
A_{23}	0.5879765	0.7903226
A_{24}	1.0674487	1.2653959
A_{25}	1.2697947	1.2697947
W	311.8757625	315.4792513

Table 23
Results for the ring structure presented by Patnaik et al. [29]

Method	Explicit gradients	Approximate gradients
SUMT	308.730	308.896
IMSL-SQP	308.587	308.729
FD	308.406	308.789
SLP	312.024	308.454

4.16.2. A 72-bar truss

A 72-bar truss structure depicted in Fig. 11 is considered now. The design variables are the cross-sectional areas of the bars and the minimum value for each one is 0.1 in.^{2}. The 72 design variables are linked in sixteen groups detailed in Table 25. The constraints involve a maximum allowable displacement of 0.25 in. at the nodes 1–16 in the directions x and y and a maximum allowable stress in each bar restricted to the range $[-25, 25]$ ksi. The density of the

Table 24
Average of penalty coefficients in the best run—ring structure

i	k_i	i	k_i	i	k_i	i	k_i	i	k_i
1	18 156.220	20	47.357	41	7123.337	60	13.725	102	3377.956
2	52.632	21	15 115.445	42	9104.164	65	101.865	104	0.33925
3	7009.447	22	16 732.545	43	12 593.860	66	0.44324	105	16 360.281
4	5745.554	23	251.73	44	5667.488	67	62.374	114	11 668.074
5	36.315	24	15 967.931	45	0.007873	68	1358.262	116	366.608
6	15 864.547	25	17 118.855	46	0.743879	69	26.791	118	6.572
7	16 090.192	26	11 520.353	47	14 457.042	70	505.054	128	0.47640
8	18.366	27	1.322	48	15 986.401	75	4.877	129	865.456
9	8556.792	28	0.372	49	11 788.493	77	1.067	130	377.3614
10	7401.741	29	4334.416	50	9.203	78	0.037492	131	2.384
11	86.056	30	8825.206	51	373.367	79	336.648	140	0.30368
12	19 219.696	31	13 123.726	52	3535.719	80	5885.583	141	1055.463
13	13 588.588	32	9669.698	53	208.604	81	1.886	142	359.546
14	184.806	33	3.943	54	10.230	82	427.970	143	1.695
15	13 967.190	35	7976.303	55	9071.179	89	1041.249	177	0.22394
16	10 824.979	36	14 775.485	56	16.363	90	4855.924	178	5946.516
17	97.021	37	15 691.882	57	171.610	91	0.011498	179	0.03132
18	12 354.704	38	6768.194	58	2808.981	93	10 303.626	182	650.041
19	12 930.074	40	24.550	59	250.858	101	3263.931		

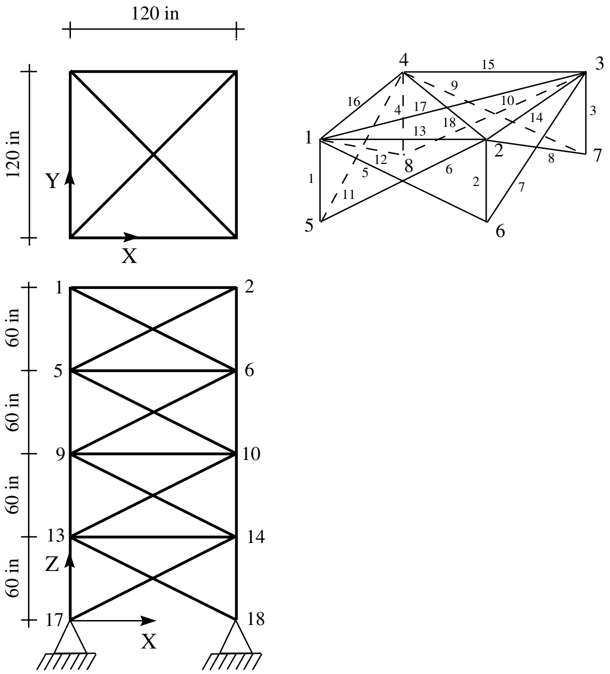

Fig. 11. The 72-bar truss example.

Table 25
Member groups of the design variables of the 72-bar truss

Group	Bars
A_1	1, 2, 3, 4
A_2	5, 6, 7, 8, 9, 10, 11, 12
A_3	13, 14, 15, 16
A_4	17, 18
A_5	19, 20, 21, 22
A_6	23, 24, 25, 26, 27, 28, 29, 30
A_7	31, 32, 33, 34
A_8	35, 36
A_9	37, 38, 39, 40
A_{10}	41, 42, 43, 44, 45, 46, 47, 48
A_{11}	49, 50, 51, 52
A_{12}	53, 54
A_{13}	55, 56, 57, 58
A_{14}	59, 60, 61, 62, 63, 64, 65, 66
A_{15}	67, 68, 69, 70
A_{16}	71, 72

Table 26
Load cases for the 72 bar-truss (kips)

Load case	Node	F_x	F_y	F_z
1	1	5	5	−5
2	1	0	0	−5
	2	0	0	−5
	3	0	0	−5
	4	0	0	−5

Table 27
Comparison of results on 72-bar truss (the optimum is not known)

	APM	CPM
Best	385.4646600	384.1341982
Average	400.0218366	402.1339046
Worst	411.4276266	401.4633786
NFE (10^4)	32	32

material is 0.1 lb/in^3 and the Young modulus is equal to 10^4 ksi. Two load cases are considered for this structure and they are defined in Table 26. The population size was set to 200 and 1600 generations were performed in 20 independent runs. 16 bits were used for each one of the 16 design variables.

Table 27 presents a comparison of results for this test-problem using the two schemes APM and CPM. A slight difference was observed in favor of the CPM scheme in the best and worst values of the objective function. Table 28 presents a comparison of results among literature and the present work with the cross-sectional area of each of the bars and the final weight reached for the 72-bar truss.

Concerning Table 28, it must be noted that, according to our computations, based on the values shown in that table, the solutions marked with an asterisk are not rigorously feasible while the one marked with a "+" (although feasible) has an actual weight of 385.6189.

5. Conclusions

A new simple adaptive parameter-less penalty scheme for the solution of constrained problems via genetic algorithms has been proposed. Its main feature, besides being adaptive and not requiring any parameter, is to automatically define a different penalty parameter for each constraint.

It has also been observed that even using the average of the adapted values for each constraint as fixed penalties from the start of a new run does not produce better results than those obtained by the proposed adaptive procedure.

Table 28
Comparison of results from the literature and the present work for the 72-bar truss

Var.	Ref. [30]	Ref. [31]	Ref. [32]$^{+}$	Ref. [33]	Ref. [34]*	Ref. [35]	Ref. [35]*	APM	CPM
A_1	0.161	0.1492	0.1641	0.1585	0.157	0.155	0.161	0.1514412	0.1534600
A_2	0.557	0.7733	0.5552	0.5936	0.537	0.535	0.544	0.5557931	0.5772519
A_3	0.377	0.4534	0.4187	0.3414	0.411	0.480	0.379	0.5012116	0.2935027
A_4	0.506	0.3417	0.5758	0.6076	0.571	0.520	0.521	0.6808072	0.7261921
A_5	0.611	0.5521	0.5327	0.2643	0.509	0.460	0.535	0.5761303	0.6547875
A_6	0.532	0.6084	0.5256	0.5480	0.522	0.530	0.535	0.4225544	0.5732891
A_7	0.100	0.1000	0.1000	0.1000	0.100	0.120	0.103	0.1000000	0.1000000
A_8	0.100	0.1000	0.1000	0.1509	0.100	0.165	0.111	0.1595163	0.1026169
A_9	1.246	1.0235	1.2893	1.1067	1.286	1.155	1.310	1.2859892	1.3158221
A_{10}	0.524	0.5421	0.5201	0.5793	0.516	0.585	0.498	0.4739208	0.4832670
A_{11}	0.100	0.1000	0.1000	0.1000	0.100	0.100	0.110	0.1000000	0.1020935
A_{12}	0.100	0.1000	0.1000	0.1000	0.100	0.100	0.103	0.1136828	0.1000748
A_{13}	1.818	1.4636	1.9173	2.0784	1.905	1.755	1.910	2.1246006	1.8623850
A_{14}	0.524	0.5207	0.5207	0.5034	0.518	0.505	0.525	0.5221469	0.4620325
A_{15}	0.100	0.1000	0.1000	0.1000	0.100	0.105	0.122	0.1000000	0.1000000
A_{16}	0.100	0.1000	0.1000	0.1000	0.100	0.155	0.103	0.1000000	0.1000000
W	381.2	395.97	379.66	388.63	380.84	385.76	383.12	385.4646600	384.1341982

W is the final weight in lb.

Table 29
Definition of each Test Problem TP

TP	Function	Constraints
1	$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$	$4.84 - (x_1 - 0.05)^2 - (x_2 - 2.5)^2 \geqslant 0$
		$x_1^2 + (x_2 - 2.5)^2 - 4.84 \geqslant 0$
2	$f(x) = 0.01x_1^2 + x_2^2$	$x_1x_2 - 25 \geqslant 0, \quad x_1^2 + x_2^2 - 25 \geqslant 0$
3	$f(x) = (x_1 - 2)^2 + (x_2 - 1)^2$	$-x_1^2 + x_2 \geqslant 0, \quad x_1 + x_2 \leqslant 2$
4	$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$	$x_1 + x_2^2 \geqslant 0, \quad x_1^2 + x_2 \geqslant 0$
5	$f(x) = -x_1 - x_2$	$x_2 - 2x_1^4 + 8x_1^3 - 8x_1^2 - 2 \leqslant 0$
		$x_2 - 4x_1^4 + 32x_1^3 - 88x_1^2 + 96x_1 - 36 \leqslant 0$
6	$f(x, \mathbf{y}) = 6.5x - 0.5x^2 - y_1 - 2y_2 - 3y_3 - 2y_4 - y_5$	$x + 2y_1 + 8y_2 + y_3 + 3y_4 + 5y_5 \leqslant 16$
		$-8x - 4y_1 - 2y_2 + 2y_3 + 4y_4 - y_5 \leqslant -1$
		$2x + 0.5y_1 + 0.2y_2 - 3y_3 - y_4 - 4y_5 \leqslant 24$
		$0.2x + 2y_1 + 0.1y_2 - 4y_3 + 2y_4 + 2y_5 \leqslant 12$
		$-0.1x - 0.5y_1 + 2y_2 + 5y_3 - 5y_4 + 3y_5 \leqslant 3$
7	$G_1(x) = 5x_1 + 5x_2 + 5x_3 + 5x_4 - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$	$2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leqslant 0$
		$2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leqslant 0$
		$2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leqslant 0$
		$-2x_4 - x_5 + x_{10} \leqslant 0, \quad -8x_1 + x_{10} \leqslant 0$
		$-2x_6 - x_7 + x_{11} \leqslant 0, \quad -8x_2 + x_{11} \leqslant 0$
		$-2x_8 - x_9 + x_{12} \leqslant 0, \quad -8x_3 + x_{12} \leqslant 0$
8	$G_3(x) = (\sqrt{n})^n \prod_{i=1}^{n} x_i$	$h(x) = \sum_{i=1}^{n} x_i^2 - 1 = 0$

Table 29 (*continued*)

TP	Function	Constraints
9	$G_4(x) = 5.3578547x_3^2 + 0.8356891x_1x_2 + 37.293239x_1 - 40\,792.141$	$0 \leqslant 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 \leqslant 92$
		$90 \leqslant 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \leqslant 110$
		$20 \leqslant 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \leqslant 25$
10	$G_5(x) = 3x_1 + 0.000001x_1^3 + 2x_2 + 0.000002/3x_2^3$	$x_4 - x_3 + 0.55 \geqslant 0, x_3 - x_4 + 0.55 \geqslant 0$
		$1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0$
		$1000\sin(x_3 - 0.25) + 1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0$
		$1000\sin(x_4 - 0.25) + 1000\sin(x_4 - x_3 - 0.25) + 1294.8 = 0$
11	$G_8(x) = \dfrac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$	$x_1^2 - x_2 + 1 \leqslant 0$
		$1 - x_1 + (x_2 - 4)^2 \leqslant 0$
12	$G_9(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2$	$127 - 2x_1^2 - 3x_2^4 - x_3 - 4x_4^2 - 5x_5 \geqslant 0$
	$+ 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$	$282 - 7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5 \geqslant 0$
		$196 - 23x_1 - x_2^2 - 6x_6^2 + 8x_7 \geqslant 0$
		$-4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^2 - 5x_6 + 11x_7 \geqslant 0$
13	$G_{11}(x) = x_1^2 + (x_2 - 1)^2$	$x_2 - x_1^2 = 0$

Actually, in one case, feasible solutions were no longer obtained, indicating the importance of the adaptive procedure. In all problems tested so far (including other applications not shown here) the procedure has produced very good results. It must also be emphasized that such good results were obtained in spite of the use of a very simple binary coded genetic algorithm.

Acknowledgements

This paper was partially written while the first author was visiting the Colorado State Artificial Intelligence Laboratory. The hospitality provided by Prof. Darrel Whitley as well as several fruitful discussions are gratefully acknowledged. The authors acknowledge the support received from CNPq (301233/86-1 and 475398/2001-7) and FAPEMIG (TEC-692/99). The authors would also like to thank the reviewers for the corrections and suggestions which helped improve the quality of the paper.

References

[1] M. Schoenauer, Z. Michalewicz, Evolutionary computation at the edge of feasibility, in: H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature—PPSN IV, vol. 1141, Springer-Verlag, LNCS, Berlin, 1996, pp. 245–254.
[2] S. Koziel, Z. Michalewicz, Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization, Evolutionary Computation 7 (1) (1999) 19–44.
[3] G.E. Liepins, W.D. Potter, A genetic algorithm approach to multiple-fault diagnosis, in: L. Davis (Ed.), Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991, pp. 237–250 (Chapter 17).
[4] D. Orvosh, L. Davis, Using a genetic algorithm to optimize problems with feasibility contraints, in: Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE Press, New York, 1994, pp. 548–553.
[5] H. Adeli, N.-T. Cheng, Augmented Lagrangian genetic algorithm for structural optimization, Journal of Aerospace Engineering 7 (1) (1994) 104–118.
[6] H.J.C. Barbosa, A coevolutionary genetic algorithm for constrained optimization problems, in: Proceedings of the Congress on Evolutionary Computation, Washington, DC, USA, 1999, pp. 1605–1611.
[7] P.D. Surry, N.J. Radcliffe, The COMOGA method: constrained optimisation by multiobjective genetic algorithms, Control and Cybernetics 26 (3) (1997).
[8] T.P. Runarsson, X. Yao, Stochastic ranking for constrained evolutionary optimization, IEEE Transactions on Evolutionary Computation 4 (3) (2000) 284–294.
[9] A.H.C. van Kampen, C.S. Strom, L.M.C. Buydens, Lethalization, penalty and repair functions for constraint handling in the genetic algorithm methodology, Chemometrics and Intelligent Laboratory Systems 34 (1996) 55–68.
[10] Z. Michalewicz, M. Schoenauer, Evolutionary algorithms for constrained parameter optimization problems, Evolutionary Computation 4 (1) (1996) 1–32.

[11] R. Hinterding, Z. Michalewicz, Your brains and my beauty: Parent matching for constrained optimization, in: Proceedings of the Fifty International Conference on Evolutionary Computation, Alaska, 4–9 May 1998, pp. 810–815.

[12] S. Koziel, Z. Michalewicz, A decoder-based evolutionary algorithm for constrained optimization problems, in: T. Bäck, A.E. Eiben, M. Schoenauer, H.-P. Schwefel (Eds.), Proceedings of the Fifth Parallel Problem Solving from Nature, Amsterdam, 27–30 September, Lecture Notes in Computer Science, Springer-Verlag, 1998.

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

[14] Z. Michalewicz, A survey of constraint handling techniques in evolutionary computation, in: Proceedings of the 4th International Conference on Evolutionary Programming, MIT Press, Cambridge, MA, 1995, pp. 135–155.

[15] Z. Michalewicz, D. Dasgupta, R.G. Le Riche, M. Schoenauer, Evolutionary algorithms for constrained engineering problems, Computers and Industrial Engineering Journal 30 (2) (1996) 851–870.

[16] R.G. Le-Riche, C. Knopf-Lenoir, R.T. Haftka, A segregated genetic algorithm for constrained structural optimization, in: L.J. Eshelman (Ed.), Proceedings of the Sixth International Conference on Genetic Algorithms, Pittsburgh, PA, 1995, pp. 558–565.

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

[18] J.A. Joines, C.R. Houck, On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GaAs, in: Z. Michalewicz, J.D. Schaffer, H.-P. Schwefel, D.B. Fogel, H. Kitano (Eds.), Proceedings of the First IEEE International Conference on Evolutionary Computation, 19–23 June, 1994, pp. 579–584.

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

[20] D.W. Coit, A.E. Smith, D.M. Tate, Adaptive penalty methods for genetic optimization of constrained combinatorial problems, INFORMS Journal on Computing 6 (2) (1996) 173–182.

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

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

[23] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, 1996.

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

[25] D. Powell, M.M. Skolnick, Using genetic algorithms in engineering design optimization with non-linear constraints, in: S. Forrest (Ed.), Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, 1993, pp. 424–430.

[26] L. Costa, P. Oliveira, Evolutionary algorithm approach to the solution of mixed integer non-linear programming problems, Computers and Chemical Engineering 25 (2001) 257–266.

[27] M.-W. Huang, J.S. Arora, Optimal design with discrete variables: Some numerical experiments, International Journal for Numerical Methods in Engineering 40 (1997) 165–188.

[28] T.J.R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Prentice Hall, Eaglewood Cliffs, NJ, 1987.

[29] S.N. Patnaik, D.A. Hopkins, R. Coroneos, Structural optimization with approximate sensitivities, Computer and Structures 58 (2) (1996) 407–418.

[30] V.B. Venkayya, Design of optimum structures, Journal of Computers and Structures 1 (1–2) (1971) 265–309.
[31] R.A. Gellatly, L. Berke, Optimal structural design, Technical Report AFFDL-TR-70-165, Air Force Flight Dynamics Laboratory, AFFDL, 1971.
[32] X. Renwei, L. Peng, Structural optimization of functions and dual theory, Computer Methods in Applied Mechanics and Engineering 65 (1987) 101–104.
[33] L.A. Schimit, B. Farshi, Some approximation concepts in structural synthesis, AIAA Journal 12 (1974) 692–699.
[34] W. Xicheng, G. Guixu, A parallel iterative algorithm for structural optimization, Computer Methods in Applied Mechanics and Engineering 96 (1993) 25–32.
[35] F. Erbatur, O. Hasançebi, I. Tütüncü, H. Kilç, Optimal design of planar and space structures with genetic algorithms, Computer and Structures 75 (2000) 209–224.