Optim Eng (2017) 18:693–722
DOI 10.1007/s11081-016-9344-z

An adaptive penalty scheme to solve constrained structural optimization problems by a Craziness based Particle Swarm Optimization

Érica C. R. Carvalho[1] · Heder S. Bernardino[1] · Patrícia H. Hallak[1] · Afonso C. C. Lemonge[1]

Received: 23 August 2015 / Revised: 26 February 2016 / Accepted: 23 October 2016 /
Published online: 28 November 2016

Abstract The use of Evolutionary Algorithms (EAs) to solve optimization problems has been increasing. One of the most used techniques is Particle Swarm Optimization (PSO), which is considered robust, efficient and competitive in comparison with other bio-inspired algorithms. EAs were originally designed to solve unconstrained optimization problems. However, the most significant problems, particularly those from real world optimization, present constraints. It is not trivial to define a strategy to handle constraints and, in general, penalty functions containing parameters to be set by the user and it may affect the search considerably. This paper consists of a combination of the Craziness based Particle Swarm Optimization (CRPSO) with an adaptive penalty technique, called Adaptive Penalty Method (APM), to solve constrained optimization problems. A CRPSO is adopted here in order to avoid premature convergence using a new velocity expression and an operator called "craziness velocity". APM and its variants were applied in other EAs, originally in a Genetic Algorithm, which demonstrated its robustness. APM deals with inequality and equality constraints, and it is free of parameters to be defined by the user. In order to assess the applicability and performance of the algorithm, several structural engineering optimization problems traditionally found in the literature are used in the computational experiments.

✉ Érica C. R. Carvalho
ericacrcarvalho@gmail.com

Heder S. Bernardino
heder@ice.ufjf.br

Patrícia H. Hallak
patricia.hallak@ufjf.edu.br

Afonso C. C. Lemonge
afonso.lemonge@ufjf.edu.br

[1] Federal University of Juiz de Fora (UFJF), Juiz de Fora, MG, Brazil

Keywords Particle swarm optimization · Adaptive penalty method · Structural optimization

1 Introduction

For years engineering fields have been developing tools to reach optimum performance designs, savings and costs. This brings significant benefits, such as the reduced use of raw material, manufacturing cost, transport, storage, and others. All these aspects should be considered in the optimization problems.

Constrained optimization can be defined as the study of problems that aims to minimize (or maximize) a function searching for the values of variables from a set of options (continuous, discrete or mixed) which satisfy the set of constraints of the problem. Increasingly, optimization has been used in a number of problems in engineering, chemistry, physics, mathematics, economics, and other areas of science.

EAs, widespread in the literature, are search algorithms which can be directly applied to unconstrained problems, searching for a set of variables that minimizes (or maximizes) one or more objective functions. Inspired by the natural processes of the real world, computational models are developed based on the concept of collective intelligence. In the nineties, studies led to the creation of a new area in computational intelligence, the swarm intelligence.

In this context, an algorithm that has been successfully applied is Particle Swarm Optimization (PSO) (Eberhart and Kennedy 1995), which is based on the simulation of the social behavior of some species of birds and fish, to solve complex problems. PSO has two main advantages: fast convergence and few parameters to be tuned, which make this technique particularly easy to implement. However, the performance of the algorithm when applied to a given problem depends on its parameters, as in many others EAs (Dobslaw 2010). In this paper a modified PSO called Craziness based Particle Swarm Optimization (CRPSO) proposed by Kar et al. (2012) is adopted in order to avoid premature convergence using a new velocity expression and an operator called "craziness velocity".

The application of EAs to constrained problems cannot be made directly. To solve this problem, the more traditional strategy used is to incorporate a penalty function into the objective function transforming a constrained problem into an unconstrained one. Notice, that other types of constraint handling techniques can be found in the literature. For instance, Monson and Seppi (2005) solves optimization problems with linear constraints using a PSO by transforming the search space into an unconstrained one.

In general, the main challenge when building penalty strategies is to create functions with less dependence on parameters set by the user, which can affect the search for the optimal solutions. For example, many works in the literature discuss techniques to handle constraints, such as Barbosa (1999), Hinterding and Michalewicz (1998), Kampen et al. (1996), Koziel and Michalewicz (1998), Koziel and Michalewicz (1999), Orvosh and Davis (1994) and Runarsson and Yao (2000) propose penalty strategies with parameters chosen by the user. Choosing the right parameters may require an exhaustive process of trial and error. On the other hand, parameter-free techniques have become more attractive to provide robustness when applied to problems of various degrees of complexity.

One of these parameterless techniques is APM (Adaptive Penalty Method), which was proposed by Barbosa and Lemonge (2002) to handle constrained optimization problems conceived for a generational Genetic Algorithm (GA). This strategy has shown to be efficient and robust (Gallet et al. 2005; Rocha and Fernandes 2009; Silva et al. 2011; Venter and Haftka 2010). APM does not require the knowledge of the explicit form of the constraints as a function of the design variables and was developed based on information obtained from the population, such as the average of objective function and the level of violation of each constraint. The idea is to observe how each constraint is being violated (or not) by the candidate solutions in the population and then to set a higher penalty coefficient for those constraints which seem to be more difficult to satisfy. In addition to the original method, Barbosa and Lemonge (2008) introduced four new variants.

Several works applying PSO to constrained optimization problems can be found in the literature. Liu and Hui (2012) proposed a new method with a PSO to solve the well-known 24 benchmark constrained optimization problems (Liang et al. 2006). Those authors developed a method called Numerical Gradient (NG) to find the feasible region. Numerical results are presented and compared with the results from the existing PSO variants dealing with constraint optimization problems.

PSO algorithm for solving constrained optimization problems was introduced by Elsayed et al. (2013). The algorithm uses two new mechanisms to guarantee that PSO will perform consistently well for all problems and will not be trapped in local optima. The first one to maintain a better balance between intensification and diversification and the second one to escape from local solutions. The performance of the proposed algorithm is analyzed by solving the CEC2010 (Tang et al. 2010) constrained optimization problems and shows consistent performance, and is superior to other state-of-the-art algorithms.

Applications of PSO in FACTS (Flexible Alternating Current Transmission System) optimization problem have been explained and analyzed by Jordehi et al. (2013). They used a basic PSO variant, parameter selection, multi-objective handling, constraint handling, and discrete variable handling. Some hints and proposals for future research in that area were provided in that paper. Mazhoud et al. (2013) presented a specific constraint-handling mechanism to solve engineering problems using an adapted particle swarm optimization algorithm. The resulting objective problem is solved using a simple lexicographic method. The new algorithm is called CVI-PSO (Constraint Violation with Interval arithmetic PSO) and the authors provide numerous experimental results based on a well-known benchmark and comparisons with previously reported results.

Innocente et al. (2015) developed a robust particle swarm algorithm coupled with a novel adaptive constraint-handling technique to search for the global optimum of management of petroleum fields aiming to increase the oil production during a given concession period of exploration. A hybrid PSO and GA, named as a PSO-GA, for solving the constrained optimization problems was present by Garg (2016).

The constraints are handled with the help of the parameter-free penalty function. The results of constrained optimization problems are superior to those reported with the typical approaches exist in the literature.

This paper aims to explore the capability of APM and some of its variants coupled to CRPSO, for the first time, as an optimization method to find solutions of constrained structural optimization problems. It is important to note that new variants, not analyzed in previous studies, are proposed in this paper.

In the next section, the general optimization problem is described. Section 3 presents a particle swarm algorithm. A brief discussion of techniques to handle constrained optimization problems is presented in Sect. 4. Numerical experiments, with several test problems from the literature, are presented in Sect. 5. Finally, in Sect. 6, the conclusions and proposed future works are presented.

2 The structural optimization problem

Typically, a structural optimization problem can be formulated in many ways. For structures made of bars (trusses, frames, etc), the constrained dimensional optimization problem consists of finding the set of areas $\mathbf{a} = \{\mathbf{A_1}, \mathbf{A_2}, \ldots, \mathbf{A_n}\}$ which minimizes the weight $W(\mathbf{a})$ of the truss structure.

$$W(\mathbf{a}) = \sum_{\mathbf{i=1}}^{\mathbf{n}} \rho \mathbf{A_i} \mathbf{L_i} \tag{1}$$

subject to the normalized displacement constraints and the normalized stress constraints

$$\frac{u_{j,k}}{\bar{u}} - 1 \leq 0, \quad 1 \leq j \leq m, \quad 1 \leq k \leq n_l \tag{2}$$

$$\frac{\sigma_{l,k}}{\bar{\sigma}} - 1 \leq 0, \quad 1 \leq l \leq n, \quad 1 \leq k \leq n_l \tag{3}$$

where ρ is the specific mass of the material, A_i and L_i are the cross-sectional area and the length of the i-th member of the structure, u_j and σ_l are, respectively, the nodal displacement of the j-th degree of freedom and the stress of the l-th member, n is the total number of members, m is the number of degree of freedom of the structure and n_l is the number of load cases to which the structure is submitted. The allowable displacements and stresses are defined as $\bar{u}$ and $\bar{\sigma}$, respectively. Additional constraints such as the minimum natural vibration frequency or buckling stress limits can also be included.

Usually, equality constraints are transformed into inequality constraints as follows

$$|h(x)| - \epsilon \leq 0 \tag{4}$$

where h is a equality constraint and ϵ is a tolerance.

3 Particle swarm optimization

Particle Swarm Optimization (PSO) was originally introduced by Eberhart and Kennedy (1995). The algorithm was inspired by the social behavior of animals such as fish schooling, insects swarming and birds flocking.

PSO is a population-based algorithm with fewer parameters to be set and is easier to implement than other EA's. PSO also shows a faster convergence rate than other EAs for solving some optimization problems (Kennedy et al. 2001). Each particle of the swarm represents a potential solution of the optimization problem. The particles fly through the search space and their positions are updated based on the best positions of individual particles in each iteration. The objective function is evaluated for each particle and the fitness value of particles is obtained in order to determine which position in the search space is the best.

In each iteration, the swarm is updated using the following equations:

$$v_j^{(i)}(t+1) = v_j^{(i)}(t) + c_1 \cdot r_1(x_{pbest}^{(i)} - x_j^{(i)}) + c_2 \cdot r_2(x_{gbest} - x_j^{(i)}) \tag{5}$$

$$x_j^{(i)}(t+1) = x_j^{(i)}(t) + v_j^{(i)}(t+1) \tag{6}$$

where $v_j^{(i)}$ and $x_j^{(i)}$ represent the current velocity and the current position of the j-th design variable of the i-th particle, respectively. $x_{pbest}^{(i)}$ is the best previous position of the i-th particle (called *pbest*) and x_{gbest} is the best global position among all the particle in the swarm (called *gbest*); c_1 and c_2 are coefficients that control the influence of cognitive and social information, respectively, and r_1 and r_2 are two uniform random sequences generated between 0 and 1.

The basic PSO algorithm can be briefly described using the following steps:

1. Initialize a random particle swarm (position) and velocity distributed within the search space.
2. Initialize x_{pbest} and x_{gbest}.
3. Calculate the objective function value of each particle of the swarm.
4. Update the position and velocity of each particle in the iteration $j+1$ using Eqs. (5) and (6).
5. Update x_{pbest} and x_{gbest}.
6. Repeat the steps 3 to 5 until a stop condition is satisfied.

A change in the conventional PSO algorithm, called CRPSO and proposed by Kar et al. (2012), is used in this study and aims to improve PSO behavior in order to avoid premature convergence. It introduces an entirely new velocity expression v_i associated with many random numbers and an operator called "craziness velocity". The operator has a predefined probability of craziness.

In this case the velocity can be expressed as follows (Kar et al. 2012):

$$v_j^{(i)}(t+1) = r_2 \cdot sign(r_3) \cdot v_j^{(i)}(t) + (1-r_2)c_1 \cdot r_1(x_{pbest}^{(i)} - x_j^{(i)}) + (1-r_2) \cdot c_2 \cdot (1-r_1)(x_{gbest} - x_j^{(i)}) + P(r_4) \cdot sign2(r_4) \cdot v_j^{craziness} \tag{7}$$

where r_1, r_2, r_3 and r_4 are the random parameters uniformly taken from the interval [0,1), $sign(r_3)$ is a function defined as

$$sign(r_3) = \begin{cases} -1, & r_3 \leq 0.05 \\ 1, & r_3 > 0.05 \end{cases} \tag{8}$$

$v_j^{craziness}$, the craziness velocity, is a user define parameter from the interval [v^{min}, v^{max}] and $P(r_4)$ and $sign2(r_4)$ are defined, respectively, as

$$P(r_4) = \begin{cases} 1, & r_4 \leq Pcr \\ 0, & r_4 > Pcr \end{cases} \tag{9}$$

$$sign2(r_4) = \begin{cases} -1, & r_4 \geq 0.5 \\ 1, & r_4 < 0.5 \end{cases} \tag{10}$$

and Pcr is a predefined probability of craziness. Although the parameter Pcr is fixed, $P(r_4)$ is defined every time the velocity is calculated.

4 Handling constraints

The constrained optimization problems are exhaustively studied in the literature. The usual approach when using evolutionary algorithms is to define strategies to handle constraints by adopting penalty functions Barbosa et al. 2015. The main idea is to transform a constrained optimization problem into an unconstrained one by adding a penalty function.

Penalty techniques can be classified as multiplicative or additive. In the multiplicative case, a positive penalty factor is introduced in order to amplify the value of the objective function of an infeasible individual in a minimization problem. In the additive case, a penalty function is added to the objective function in order to define the fitness value of an infeasible individual. They can be further divided into interior and exterior techniques (Barbosa et al. 2015). The idea in both cases is to amplify the value of the fitness function of an infeasible individual.

Penalty methods can be also classified as static, dynamic and adaptive. Static penalty depends on the definition of an external factor to be added to or multiplied by the objective function. Dynamic penalty methods, in general, have penalty coefficients directly related to the number of generations, and the adaptive penalty considers the level of violation of the population by constraints during the evolutionary process. This paper does not attempt to cover the current literature on constraint handling and the reader is referred to survey papers or book chapters of e.g. Barbosa et al. (2015), Coello (2002), Mezura-Montes and Coello (2011), Michalewicz (1995) and Michalewicz and Schoenauer (1996).

An adaptive penalty method (APM) was originally introduced by Barbosa and Lemonge (2002) for application in constrained optimization problems. That method does not require any type of user defined penalty parameter and uses information from the population, such as the average of the objective function and the level of violation of each constraint during the evolution. The fitness function is written as

$$f(\mathbf{x}) = \{\mathbf{f}(\mathbf{x}) = \{\, \mathbf{f}(\mathbf{x}), \quad \textbf{if x is feasible } \mathbf{f}(\mathbf{x}) + \textstyle\sum_{\mathbf{j=1}}^{\mathbf{m}} \mathbf{k_j v_j}(\mathbf{x}), \quad \textbf{otherwise} \;. \tag{11}$$

where

$$\bar{f}(x) = \begin{cases} f(\mathbf{x}), & \text{if} \quad f(\mathbf{x}) > \langle f(\mathbf{x}) \rangle \\ \langle f(\mathbf{x}) \rangle, & \text{if} \quad f(\mathbf{x}) \leq \langle f(\mathbf{x}) \rangle \end{cases} \tag{12}$$

and $\langle f(\mathbf{x}) \rangle$ is the average of the objective function values in the current population.

The penalty parameter k_j is defined at each generation by

$$k_j = |\langle f(\mathbf{x}) \rangle| \frac{\langle \delta_j(\mathbf{x}) \rangle}{\sum_{l=1}^{m} [\langle \delta_l(\mathbf{x}) \rangle]^2} \tag{13}$$

where $\langle \delta_j(\mathbf{x}) \rangle$ is the violation of the jth constraint averaged over the current population.

Barbosa and Lemonge (2008) presented 4 variants for APM, written as

- Variant 1: the method (1) computes the constraint violation v_j in the current population, (2) updates all penalty coefficients k_j and (3) keeps them fixed for a number of generations.
- Variant 2: the method (1) accumulates the constraint violations for t generations, (2) updates the penalty coefficients k_j and (3) keeps the penalty coefficients k_j fixed for t generations.
- Variant 3: no penalty coefficient k_j is allowed to have its value reduced along the evolutionary process, so $k_j^{new} < k_j^{current}$ then $k_j^{new} = k_j^{current}$;
- Variant 4: the method is defined as $k_j^{new} = \theta k_j^{new} + (1-\theta) k_j^{current}$, where $\theta \in [0, 1]$.

New variants for APM have been proposed and the main variations occur in the calculation of $\bar{f}$ and the calculation of the penalty coefficient k_j. A combination of these two variations with some characteristics of the variants proposed by Barbosa and Lemonge (2008) generated three new variants and they are outlined as follows

- Variant 5: $\bar{f}(x)$ is modified as follows

$$\bar{f}(x) = \begin{cases} f(\mathbf{x}), & \text{if} \quad f(\mathbf{x}) > \lfloor f(\mathbf{x}) \rfloor \\ \lfloor f(\mathbf{x}) \rfloor, & \text{otherwise.} \end{cases} \tag{14}$$

  where $\lfloor f(\mathbf{x}) \rfloor$ is the value of the objective function of the worst feasible individual. The average of the objective function is used when no feasible individual exist.

- Variant 6: $\langle \delta_j(\mathbf{x}) \rangle$, which originally represented the average of the violations of all individuals at each constraint, is defined here as the sum of the violation of all individuals which violate the j-th constraint divided by the number of individuals which violate this constraint.
- Variant 7: $\langle \delta_j(\mathbf{x}) \rangle$, which originally represented the average of the violations of all individuals at each constraint, is defined here as the sum of the violation of all individuals which violate the j-th constraint divided by the number of individuals which violate this constraint. And $\langle f(\mathbf{x}) \rangle$, which represented the average of the objective function, now denoted by $\langle\langle f(\mathbf{x}) \rangle\rangle$, is the sum of the objective function of all individuals of the current population divided by the number the infeasible individuals, given by Eq. (15).

$$k_j = |\langle\langle f(\mathbf{x}) \rangle\rangle| \frac{\langle \delta_j(\mathbf{x}) \rangle}{\sum_{l=1}^{m} [\langle \delta_l(\mathbf{x}) \rangle]^2} \tag{15}$$

5 Numerical experiments

In this section, two set of numerical experiments are performed. The parameters of CRPSO are: c_1, c_2, $v_j^{craziness}$ and Pcr.

The choice of these parameters was based on the reference Kar et al. (2012). The values for c_1 and c_2 were defined as 2.05 and $v_j^{craziness}$ and Pcr in the reference are 0.0001 and 0.3, respectively. Based on these values, firstly some test were made leading to $v_j^{craziness} = 0.001$ and $Pcr = 0.5$.

5.1 First set of numerical experiments

In order to assess the performance of the proposed algorithm, firstly, nine examples taken from the optimization literature are used. The examples are: Pressure Vessel (Mezura-Montes et al. 2003), Welded Beam (Mezura-Montes et al. 2003), Tension/Compression Spring (Mezura-Montes et al. 2003), 3 of 24 test problems known as G-Suite (Liang et al. 2006): G04 (P4), G06 (P2) and G09 (P3), and 3 problems taken from Parsopoulos and Vrahatis (2002): Problem 1 (P1), Problem 5 (P5) and Problem 6 (P6).

In all experiments the swarm size is 20 and the maximum generations is 10000. The results presented in Hu et al. (2003) and Parsopoulos and Vrahatis (2002) are compared to those obtained here in the preliminary experiments. Hu et al. (2003) proposed a modified PSO for engineering optimization problems with constraints. Parsopoulos and Vrahatis (2002) also presented a PSO algorithm in coping with constrained optimization problems.

Table 1 presents the results for the three first experiments. It can be observed that "This study" achieved similar or higher results when compared to the reference. Table 2 presents the results for the Problems 1 to 7. The results for these

Table 1 Comparison of the results of the minimization of the weight of a pressure vessel, welded beam and tension/compression spring

Pressure vessel			Welded beam			Tension/compression spring		
	This study	Hu et al. (2003)		This study	Hu et al. (2003)		This study	Hu et al. (2003)
T_s	0.8125	0.8125	x_1	0.182873	0.20573	d	0.050072	0.051466369
T_h	0.4375	0.4375	x_2	3.805085	3.47049	D	0.319052	0.351383949
R	42.09844	42.09845	x_3	9.584571	9.03662	N	13.895301	11.60865920
L	176.6366	176.6366	x_4	0.182880	0.20573	g_1	−0.000036	−0.003336613
g_1	0.0	0.0	g_1	−2.668937	0.0	g_2	−0.000024	−1.0970128e-4
g_2	−0.035881	−0.35881	g_2	−0.207360	0.0	g_3	−3.972002	−4.0263180998
g_3	−0.327176	−0.032551	g_3	−0.000006	−5.55111e-17	g_4	−0.753919	−0.7312393333
g_4	−63.36340	−63.363329	g_4	−1.452013	−3.342983			
			g_5	−0.236367	−0.080729			
			g_6	−3.495020	−0.235540			
			g_7	−0.057874	−9.09494e-13			
W	6059.714564	6059.1311296		1.642056	1.72485084		0.012715	0.0126661409

experiments are competitive when compared with those obtained from the literature. It is important to note that the sums of constraint violations are zero in all the results found in Table 2 for "This study".

A last example (P7) in this first set of experiments is G11 (Liang et al. 2006):

$$\min f(x) = x_1^2 + (x_2 - 1)^2$$
$$\text{subject to } h(x) = x_2 - x_1^2 = 0$$

where $-1 \leq x_1 \leq 1$ and $-1 \leq x_2 \leq 1$. The optimum solution is $f(x^*) = 0.7499$.

A total of 60 particles and a budged of 300000 objective function evaluations were used. The results of 30 independent runs are shown in the last row of Table 2. The results presented in Wang and Cai (2009) are used in the comparisons. The equality constraints were transformed to inequality constraints (Eq. 4) using a tolerance equal to 10^{-4}.

Finally, the fitness values of the best solutions achieved in a given independent run are plotted in Figs. 1, 2, and 3. These figures illustrate the convergence observed, respectively, in the problems Tension/Compression Spring, Problem 1 and Problem 6 when using CRPSO. Figure 1a shown that the best solution found when 10,000 iterations are allowed is 0.012715; this fitness value is slightly smaller (0.07%) than that obtained at iteration 100, 0.012724 (Fig. 1b). Thus, one can see that CRPSO is able to produce good solutions with few objective function evaluations.

Table 2 Comparison of the results of the minimization of the weight of problems 1 to 7

Problem	Method	Average (sum viol.)	SD	Best (sum viol.)
1	This study	1.3957 (0.0)	0.00514	1.3941 (0.0)
	PSO-Bo (Parsopoulos and Vrahatis 2002)	1.3934 (0.00002)	0.0	1.3934 (0.00002)
2	This study	−6577.6340 (0.0)	139.3399	−6676.1622 (0.0)
	PSO-Bo (Parsopoulos and Vrahatis 2002)	−6961.7740 (0.000013)	0.14	−6961.8370 (0.000019)
3	This study	741.8643 (0.0)	229.7471	694.3282 (0.0)
	PSO-Bo (Parsopoulos and Vrahatis 2002)	680.6830(0.000015)	0.041	680.6360 (0.0)
4	This study	−30588.8122 (0.0)	199.5556	−30653.8756 (0.0)
	PSO-Bo (Parsopoulos and Vrahatis 2002)	−31493.1900 (1.331)	131.67	−31544.4590 (1.311)
5	This study	−31026.1347 (0.0)	1.37	−31026.4258(0.0)
	PSO-Bo (Parsopoulos and Vrahatis 2002)	−31525.4920 (0.968)	23.392	−31545.0540 (0.999)
6	This study	−213.0 (0.0)	0.0	−213.0 (0.0)
	PSO-Bo (Parsopoulos and Vrahatis 2002)	−213.0 (0.0)	0.0	−213.0 (0.0)
7	This study	0.7522 (0.0)	0.0397	0.7499 (0.0)
	Wang and Cai (2009)	0.7499 (0.0)	0.0000	0.7499 (0.0)

Sum viol. means sums of violated constraints

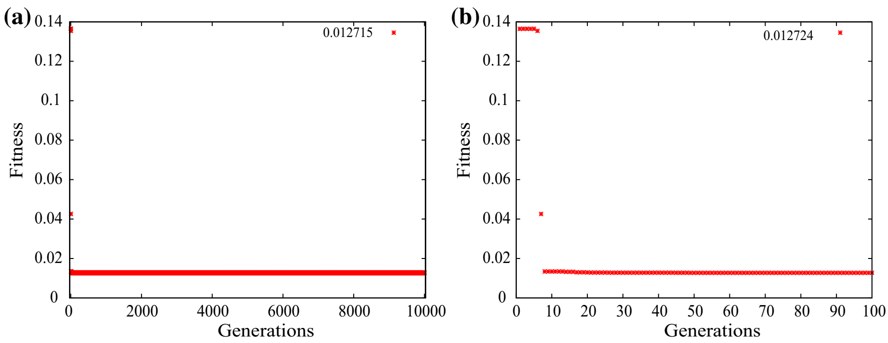


Fig. 1 Convergence for the tension/compression spring. **a** Using 10000 generations (best = 0.012715) and **b** using 100 generations (best = 0.012724)

Figures 2 and 3 indicate the same behavior with respect to Problems 1 and 6. In Fig. 2a, the best solution found for the Problem 1 at iteration 10000 is 1.3941, which is the same value obtained at iteration 100 (as observed in Fig. 2b). When analyzing the results achieved in Problem 6 (Fig. 3), the best solution found at iteration 10000 and 100 were −213.000.

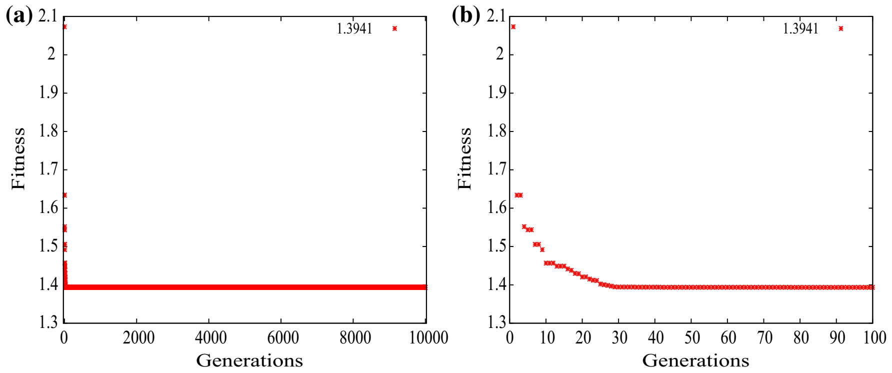


Fig. 2 Convergence for the Problem 1. **a** using 10,000 generations (best = 1.3941) and **b** using 100 generations (best = 1.3941)

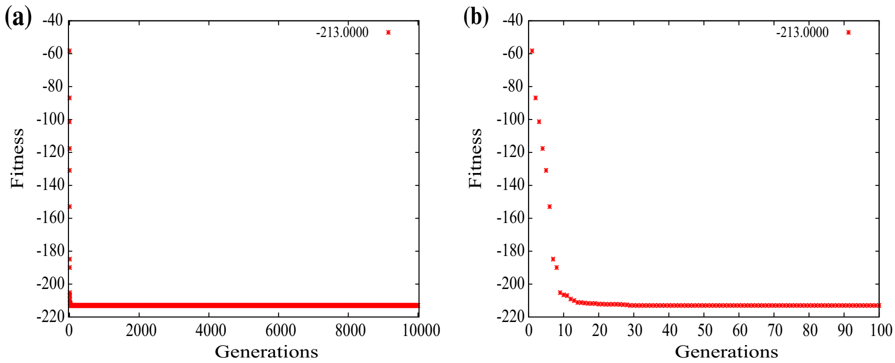


Fig. 3 Convergence for the Problem 6. **a** using 10,000 generations (best = -231.0000) and **b** using 100 generations (best = -213.0000)

These results encourage the use of CRPSO in a second set of numerical experiments presenting more complexity in terms of the number of constraints and in this case they are implicit.

5.2 Second set of numerical experiments

Several engineering problems, commonly analyzed in the literature, were addressed here. The problems correspond to 5 sizing structural optimization of plane and spatial trusses: 10-, 25-, 52-, 60- and 72-bar trusses, displayed in Figs. 4, 5, 6, 7 and 8. All these truss structures are analyzed by the Finite Element Method (FEM) during the evolutionary process.

The standard mono-objective sizing structural optimization problem, in the second set of numerical experiments, reads: find the set of cross-sectional areas of the bars $x = \{A_1, A_2, \ldots, A_N\}$ which minimizes the weight W of the structure

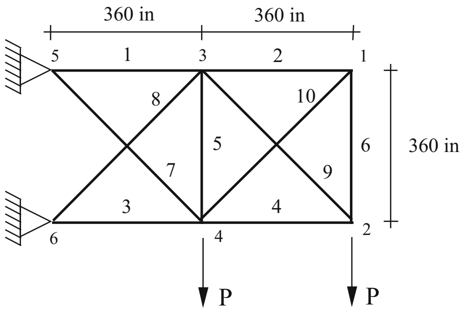


Fig. 4 10-bar truss, taken from Lemonge and Barbosa (2004)

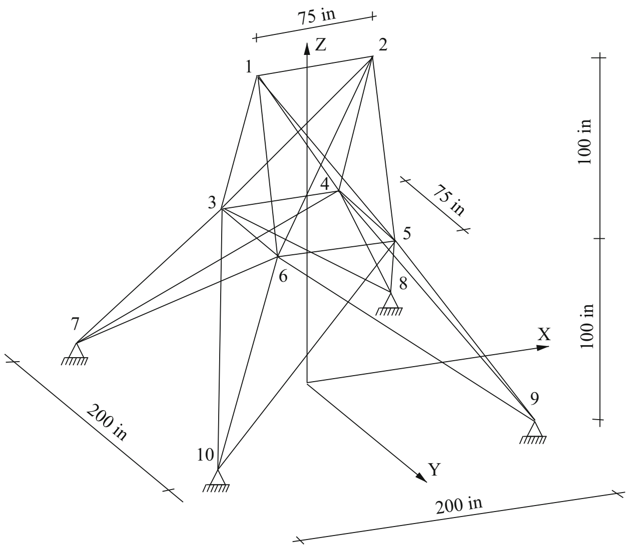


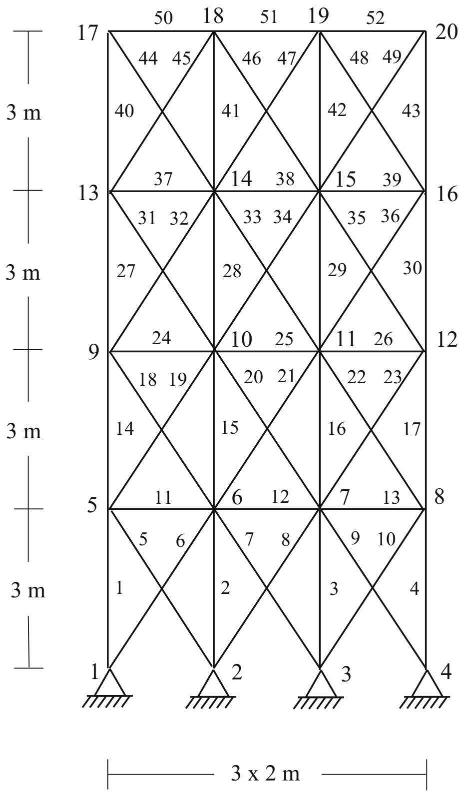
Fig. 5 25-bar truss, taken from Lemonge and Barbosa (2004)

$$W = \sum_{i=1}^{N} \rho_i A_i l_i, \tag{16}$$

where ρ_i is the density of the material, l_i is the length of the i-th bar of the truss and N is the number of bar. The problem is usually subject to inequality constraints $g_p(x) \le 0$, $p = 1, 2, \ldots, \bar{p}$.

Table 3 presents some details of each test problem. In this table, *ndv* and *type* mean the number and type of design variable (c—continuous or d - discrete), respectively, *nc* is the number of constraints and *nfe* means the number of function evaluations. Test Problem 1 has the discrete case and TP2 has the continuous case. In this table there are references where it is possible to obtain the complete data for each structural optimization problem.

Fig. 6 52-bar truss, taken from Lemonge and Barbosa (2004)



In all experiments the swarm size is 50. It should be understood that continuous design variables are considered as the nearest integer of the corresponding variable of the vector solution (particle), as used for Vargas et al. (2015). It points to either an index of a table of discrete values or an integer. In all analyses feasible solutions were found in the 35 independents runs. A preliminary analysis involving all the APM variants, given in Sect. 4, is proposed in Sect. 5.4 in order to find those that obtained the best performance in all experiments. Subsequently, only the best ones are used in the comparisons.

5.3 Performance analysis of experiments

The experiments are compared using the performance profiles proposed by Dolan and Moré (2002). Performance profiles are proposed as an analytical resource for the visualization and interpretation of the results of numerical experiments in order to define which algorithm provides a better performance considering a given set of problems.

Consider the set P of test problems p_j, with $j = 1, 2, \ldots, n_p$, a set of algorithm a_i with $i = 1, 2, \ldots, n_a$ and $t_{p,a} > 0$ a performance metric (for example, computational time, average, etc.). The performance ratio is defined as:

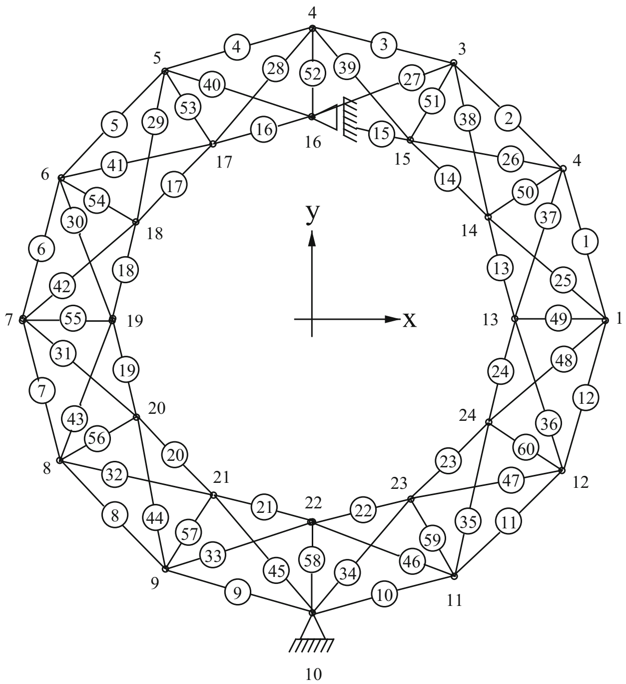


Fig. 7 60-bar truss, taken from Barbosa and Lemonge (2003)

$$r_{p,a} = \frac{t_{p,a}}{min\{t_{p,a} : a \in A\}.} \tag{17}$$

Thus, the performance profile of the algorithm is defined as:

$$\rho_a(\tau) = \frac{1}{n_p} \left| \{ p \in P : r_{p,a} \leq \tau \} \right| \tag{18}$$

where $\rho_a(\tau)$ is the probability that the performance ratio $r_{p,a}$ of algorithm $a \in A$ is within a factor $\tau > \, = 1$ of the best possible ratio. If the set P is representative of problems yet to be addressed, then algorithms with larger $p_a(\tau)$ are to be preferred. The performance profiles have a number of useful properties (Barbosa et al. 2010; Dolan and Moré 2002). Other analyses using performance profiles in algorithm performance comparison can be found in Barbosa et al. (2010) and Bernardino et al. (2011).

5.4 Analysis of APM variants

Performance profiles, introduced in the previous section, are used as a tool for a more detailed and conclusive analysis of the large volume of results.

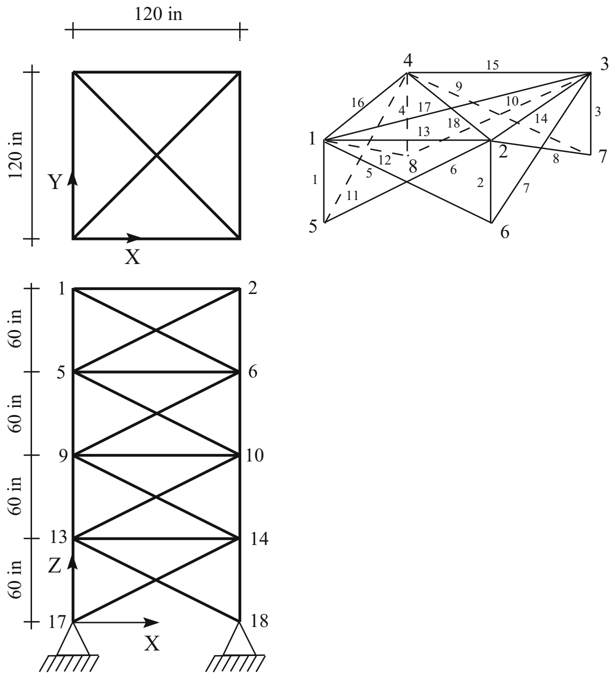


Fig. 8 72-bar truss, taken from Lemonge and Barbosa (2004)

Table 3 Second set of numerical experiments

Problem	Reference	*ndv*	*type*	*nc*	*nfe*
TP1 10-bar truss	Gellatly and Berke (1971)	10	d	18	90,000
TP2 10-bar truss	Gellatly and Berke (1971)	10	c	18	280,000
TP3 25-bar truss	Rajeev and Krishnamoorthy (1992)	8	d	43	20,000
TP4 52-bar truss	Wu and Chow (1995)	12	d	52	17,500
TP5 60-bar truss	Patnaik et al. (1996)	25	c	198	12,000
TP6 72-bar truss	Venkayya (1971)	16	c	240	35,000

Figure 9a, in which $\tau \in [1; 1.00009]$, shows the performance profiles curves, where APM presented the highest value of $\rho_a(1)$, which means that the method had the best performance considering all problems. Figure 9b shows the performance profiles where the Variant 7 had the lowest value of τ, such that $\rho_a(\tau) = 1$, and, therefore, this is considered the most robust variant. Table 4 shows the areas under the curves of the normalized performance profiles, where Variant 7, Variant 5, APM and Variant 3 had the highest area values and they are considered the methods with the best global performance. Therefore, these methods will be used in a later comparison of results.

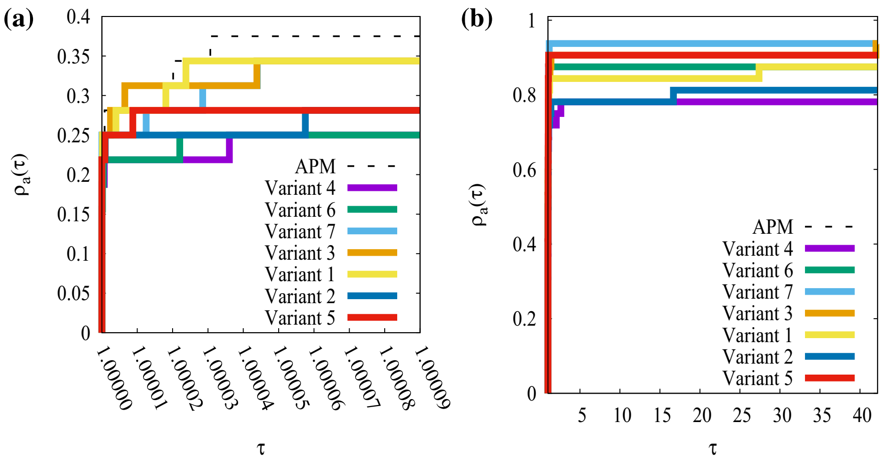


Fig. 9 Performance profiles of the variants used

Table 4 Normalized areas under the curves of the performance profiles in Fig. 9 for each variant

Variant	7	5	APM	3	6	1	2	4
Area	**1**	**0. 9668**	**0. 9664**	**0. 9663**	0.9333	0.9118	0.8536	0.8313

It is worth remembering that not all curves come in $\rho_a(\tau) = 1$. This is because for this comparison a suite composed of 35 test problems was used: 24 test functions known as G-Suite (Liang et al. 2006), 5 mechanical engineering problems: Tension/Compression Spring, Speed Reducer, Welded Beam, Pressure vessel (Mezura-Montes et al. 2003) and Cantilever beam (Erbatur and Hasançebi 2000) and 6 structural engineering problems set out in this paper in the beginning of Sect. 5. Thus, as shown in Fig. 9, feasible results for all analyzed problems were not obtained.

5.5 Results

In all experiments, 35 independent runs were performed and only the feasible solutions were considered. The results are presented in Tables 5, 6, 7, 8, 9, 10, 11, 12, 13 and 14 and the best ones are displayed in **boldface**. The tables present the *method*, the *best* solutions, the values of *median*, *average* and the *std* (standard deviation) and, finally, the *worst* solutions.

Table 5 presents the results for the 10-bar truss (discrete and continuous case) and the 25-bar truss. For the 10-bar truss (discrete case), the four methods analyzed achieved the same final weight of 5509.7173 lbs. However, APM has the best value of the median 5528.0869 lbs and Variant 5 has the best average value and the lowest value for the standard deviation, 5624.9286 lbs and 1.3148e + 03, respectively. For the continuous case, Variant 5 has the best value for the final weight of 5060.9067

Table 5 Results for the 10-bar truss (discrete and continuous case) and the 25-bar truss

	Method	Best	Median	Average	std	Worst
TP1	APM	**5509.7173**	**5528.0869**	5628.8689	1.4468e+03	6593.1205
	Variant 3	**5509.7173**	5540.4065	5700.1346	1.9561e+03	6654.2616
	Variant 5	**5509.7173**	5532.1210	**5624.9286**	**1.3148e+03**	**6540.4413**
	Variant 7	**5509.7173**	5626.0477	5665.151541	1.5179e+ 03	**6540.4413**
TP2	APM	5060.9524	5076.9778	5076.1883	2.7776e+02	5314.8520
	Variant 3	5060.9156	5076.8255	5077.9349	2.9670e+02	5191.0171
	Variant 5	**5060.9067**	5073.3360	5073.2535	2.2178e+02	5191.4245
	Variant 7	5060.9795	**5063.8619**	**5071.2579**	**1.8799e+02**	**5190.9494**
TP3	APM	**484.8541**	**485.0487**	485.97338	1.3601e+01	496.2520
	Variant 3	**484.8541**	**485.0487**	485.5626	7.2751e+00	490.9124
	Variant 5	**484.8541**	**485.0487**	**485.4513**	**6.2653e+00**	**490.1923**
	Variant 7	**484.8541**	**485.0487**	487.22217	4.1581e+01	526.8441

Table 6 Results for the 52-, 60- and 72-bar trusses

	Method	Best	Median	Average	SD	Worst
TP4	APM	**1902.6058**	**1904.1270**	1939.4710	5.2820e+02	2234.5831
	Variant 3	**1902.6058**	1910.9423	1971.9014	6.7235e+02	2340.1034
	Variant 5	**1902.6058**	1907.9052	1997.9600	1.4390e+03	2922.7265
	Variant 7	**1902.6058**	1907.9356	**1933.2624**	**3.7283e+02**	**2159.2544**
TP5	APM	312.2398	330.5434	337.5656	1.8345e+02	466.6698
	Variant 3	**311.9716**	**319.6148**	**330.9008**	**1.2286e+02**	**390.5690**
	Variant 5	313.4126	323.4483	332.0948	1.2413e+02	409.9210
	Variant 7	312.5879	334.0380	344.6552	1.8445e+02	439.2875
TP6	APM	379.65373	379.74019	383.87949	1.0493e+02	475.87770
	Variant 3	379.65592	379.72484	**381.10749**	**4.7899e+01**	**428.31708**
	Variant 5	379.65388	**379.72401**	383.86117	1.0496e+02	475.91189
	Variant 7	**379.65356**	379.72416	394.40452	2.3324e+02	545.23741

lbs and the best value for median, average, standard deviation and worst are found by Variant 7.

For the 25-bar truss, the four methods had the same weight of 484.8541 kg and the same median value of 485.0487 lbs. The best average and the lowest standard deviation was found by Variant 5, which stands out above the other methods for this problem.

The results for the 52-, 60- and 72-bar trusses are presented in Table 6. For the 52-bar truss all methods reached the same final weight of 1902.6058 kg. APM presented the best median 1904.1270 kg and Variant 7 presented the best average, the smallest standard deviation and the lowest worst value, and, thus, was the method that stood out among the methods analyzed for this problem.

For the 60-bar, the method which obtained the best performance was Variant 3, with a final weight of 311.9716 lbs and median of 319.6148 lbs. Finally, for the 72-bar truss, Variant 7 obtained the best final weight, 379.6535 lbs. However, the other methods also presented similar values. Variant 5 presented the best median value and Variant 3 is the variant with the best average and the smallest standard deviation.

5.6 Analysis of the results

The **best** and **average** values obtained by the variants are used here as performance metrics. The choice among these values depends on what the designer wants. The first analysis using the value of the best, Fig. 10a, in the range $\tau \in [1; 1.00005]$, shows the performance profile graphic, where Variant 7 presented the highest value of $\rho_a(1)$ and, therefore, had the best performance in a larger number of problems. In Fig. 10b, all variants had the lowest value of τ, such that $\rho_a(\tau) = 1$. However, Variant 3 was the method which presented the highest robustness. The areas under the performance profile curves are shown in Table 7. Variant 3, followed by Variant 7, obtained the highest values, 1 and 0.92846, respectively. Thus, Variant 3 should be considered the method with the best global performance using the best value.

The performance profiles using the average values show interesting behavior among the variants analyzed. It can be observed that Variant 7 is the variant with better performance in the majority of problems (Fig. 11a), but it is not considered robust. On the other hand, APM and Variant 3 do not present the best performance in the majority of problems but presented the best performance in general (more robustness) (Fig. 11b).

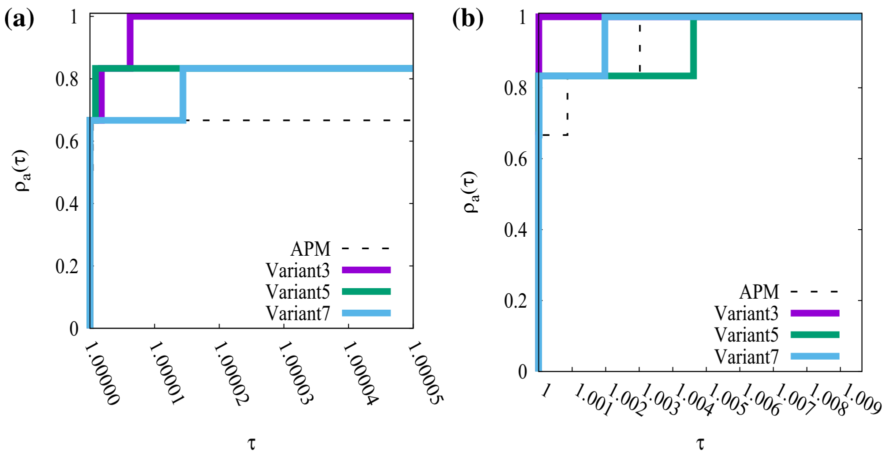


Fig. 10 Performance profiles using the **best** value

Table 7 Normalized areas under the performance profiles curves using the **best** value

Method	Variant 3	Variant 7	APM	Variant 5
Area	1	0.92846	0.86026	0.83354

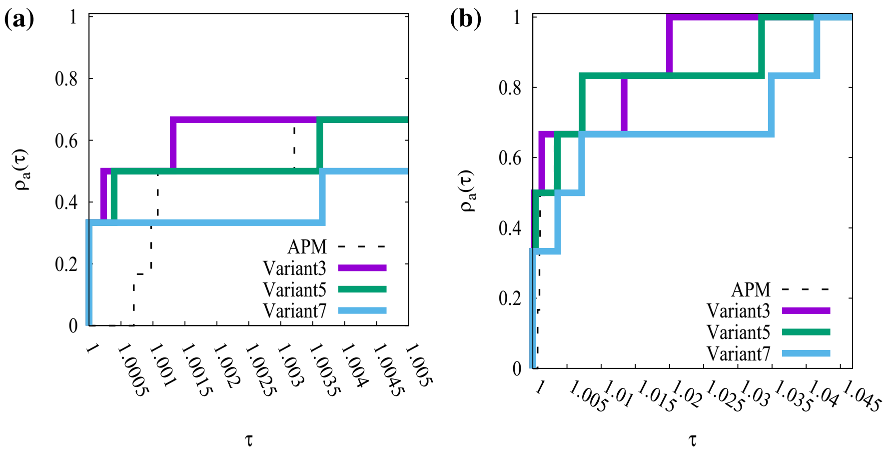


Fig. 11 Performance profiles using the **average** value

Table 8 Normalized areas under the performance profiles curves using the **average** value

Method	APM	Variant 3	Variant 5	Variant 7
Area	1	0.99292	0.94760	0.75057

In summary, Variant 3 is the variant that performed better when the best value of the objective function is considered, followed by Variant 7. Using the average value as the performance metric, the original APM found the best result among the variants analyzed, followed by Variant 3. Thus, considering the best value as performance metric commonly used among designers, Variant 3 is the variant with the best global performance among the set of methods analysed.

Tables 9, 10, 11, 12, 13 and 14 show the final design variables, the objective function (W) and the number of function evaluations (*nfe*) for the Test Problems 1–6 using Variant 3 compared with results found in the literature. The stop criteria adopted here is the maximum number of objective function evaluations (indicated by *nfe*). The values presented in Tables 9, 10, 11, 12, 13 and 14 indicate that Variant 3 obtained results competitive with those found in the literature. One can notice that the results obtained by the proposed technique are: (1) better than the result presented in Lemonge et al. (2015) and equal to those from the other references, when using the same number of function evaluations and when TP3 (Table 11) is solved; (2) equal to that presented in Sadollah et al. (2012) and better than those

Table 9 Final design variables and objective function (weight—W) found for the 10-bar truss - discrete case (TP1)

Variables	This study	SSGA (Lemonge et al. 2015)	DE (Ho-Huu et al. 2016)	aeDE (Ho-Huu et al. 2016)	MBA (Sadollah et al. 2012)
1	33.500	33.500	33.500	33.500	30.000
2	1.620	1.620	1.620	1.620	1.620
3	22.000	22.900	22.900	22.900	22.900
4	16.900	14.200	14.200	14.200	16.900
5	1.620	1.620	1.620	1.620	1.620
6	1.620	1.620	1.620	1.620	1.620
7	7.970	7.970	7.970	7.970	7.970
8	22.000	22.900	22.900	22.900	22.900
9	22.000	22.000	22.000	22.000	22.900
10	1.620	1.620	1.620	1.620	1.620
W	5509.717	**5490.738**	**5490.738**	**5490.738**	5507.750
nfe	20000	40000	6440	2380	3600

Table 10 Final design variables and objective function (weight—W) found for the 10-bar truss—continuous case (TP2)

Variables	This study	SSGA (Lemonge et al. 2015)	TCELL (Aragón et al. 2010)	WEO (Kaveh and Bakhshpoori 2016)	IMCSS (Kaveh et al. 2015)
1	30.53823	30.46297	31.23829	30.57550	30.02580
2	0.10000	0.10000	0.31662	0.10000	0.10000
3	23.06187	23.28281	23.61073	23.33680	23.62770
4	15.26055	15.20222	14.50669	15.14970	15.97340
5	0.10000	0.10000	0.31623	0.10000	0.10000
6	0.54001	0.54515	0.31623	0.52760	0.51670
7	7.46597	7.44674	8.13509	7.44580	7.45670
8	21.02079	21.02321	21.61828	20.98920	21.43740
9	21.60389	21.55451	21.22159	21.52360	20.74430
10	0.10000	0.10000	0.31634	0.10000	0.10000
W	5060.915	**5060.875**	5142.30	5060.990	5064.600
nfe	280000	80000	280000	19540	8475

obtained by the other references, when TP4 (Table 12) is considered; (3) better than those presented by the references used in the comparisons, when solving TP6 (Table 14); and (4) worse, but competitive, than the results found in the literature, when TP1, TP2, and TP5 are considered.

Figure 12 plots the fitness value of the best solution found in every iteration for the 72-bar truss. In Fig. 12a, one can seen that the best solution found at iteration

Table 11 Final design variables and objective function (weight—W) found for the 25-bar truss (TP3)

Variables	This study	SSGA (Lemonge et al. 2015)	DE (Ho-Huu et al. 2016)	aeDE (Ho-Huu et al. 2016)	MBA (Sadollah et al. 2012)
1	0.100	0.100	0.100	0.100	0.100
2	0.300	0.500	0.300	0.300	0.300
3	3.400	3.400	3.400	3.400	3.400
4	0.100	0.100	0.100	0.100	0.100
5	2.100	2.600	2.100	2.100	2.100
6	1.000	0.900	1.000	1.000	1.000
7	0.500	0.400	0.500	0.500	0.500
8	3.400	3.400	3.400	3.400	3.400
W	**484.854**	486.497	**484.854**	**484.854**	**484.854**
nfe	20000	20000	3500	1440	2150

Table 12 Final design variables and objective function (weight—W) found for the 52-bar truss (TP4)

Variables	This study	SSGA (Lemonge et al. 2015)	DE (Ho-Huu et al. 2016)	aeDE (Ho-Huu et al. 2016)	MBA (Sadollah et al. 2012)
1	4658.055	4658.055	4658.055	4658.055	4658.055
2	1161.288	1161.288	1161.288	1161.288	1161.288
3	494.193	285.161	494.193	494.193	494.193
4	3303.219	3303.219	3303.219	3303.219	3303.219
5	940.000	940.000	940.000	940.000	940.000
6	494.193	645.160	363.225	641.289	494.193
7	2238.705	2238.705	2238.705	2238.705	2238.705
8	1008.385	1008.385	1008.385	1008.385	1008.385
9	494.193	641.289	641.289	363.225	494.193
10	1283.868	1283.868	1283.868	1283.868	1283.868
11	1161.288	1161.288	1161.288	1161.288	1161.288
12	494.193	506.451	494.193	494.193	494.193
W	**1902.605**	1907.383	1903.366	1903.366	**1902.605**
nfe	17500	20000	13240	3720	5450

700 is 379.655920. When 70 iterations are considered, the value obtained is 387.312588 (Fig. 12b).

New comparisons were provided in this paper. Plots were generated using the values of normalized constraints, obtained from the solutions displayed in Tables 9, 10, 11, 12, 13 and 14. The constraints in the analyzed structural optimization problems presented in this paper are given by the Eq. (2) ($u_{j,k}/\bar{u} - 1 \leq 0$, node displacements) and Eq. (3) ($\sigma_{l,k}/\bar{\sigma} - 1 \leq 0$, normal stress). It can be expected, a

Table 13 Final design variables and objective function (weight—W) found for the 60-bar truss (TP5)

Variables	This study	SSGA (Lemonge et al. 2015)	APM (Barbosa and Lemonge 2003)	CPM (Barbosa and Lemonge 2003)	CP (Farshi and Alinia-Ziazi 2010)	IGATA (Li et al. 2014)
1	1.171786	1.230719	1.120234	1.190615	2.027300	1.218000
2	2.120145	2.186131	2.021994	2.277126	0.500000	2.161900
3	0.500000	0.500000	0.508797	0.605571	1.778100	0.000000
4	1.710471	1.811974	1.727272	1.573313	1.777500	1.769100
5	1.695954	1.402465	1.520527	1.375366	0.579300	1.738100
6	0.543523	0.550943	0.526393	0.508797	1.830500	0.000000
7	2.038263	1.931011	1.903225	1.934017	1.794700	1.975000
8	1.895264	2.259442	2.127566	2.052785	0.983000	2.079100
9	1.051748	1.057859	0.988269	1.239002	1.903100	2.053900
10	1.726629	1.926914	2.052785	1.819648	1.949700	1.786000
11	1.739545	1.663640	2.052785	1.639296	0.500000	1.769200
12	0.503238	0.500000	0.724340	0.526393	2.013500	0.000000
13	2.195972	2.502185	1.960410	2.197947	1.244100	2.161800
14	1.271449	1.247582	1.230205	1.234604	1.015600	1.248400
15	1.041964	1.043955	0.997067	1.049853	0.689600	1.082100
16	0.567591	0.542620	0.605571	0.759530	0.723300	0.696900
17	0.695008	0.500083	0.728739	0.614369	1.057800	0.714900
18	1.214659	1.065114	1.093841	1.120234	1.122600	0.996900
19	1.148367	1.120449	1.115835	1.115835	1.151200	1.157900
20	1.156885	1.149740	1.168621	1.155425	1.066400	1.141300
21	1.186998	1.072212	1.067448	1.186217	1.046700	1.023000
22	1.082519	1.124664	1.063049	1.071847	0.703900	1.195500
23	0.584492	0.666752	0.587976	0.790322	1.028000	0.697000
24	1.079818	1.121616	1.067448	1.265395	1.258800	1.082200
25	1.304250	1.556643	1.269794	1.269794	1.147500	1.248400
W	311.971	317.047	311.875	315.479	**308.590**	309.843
nfe	12000	60000	800000	80000	45	2500

value equal to “0” for active constraints. The desirable solutions are those where the constraints are active, i.e., equal to “0” or very close to it. The plots of Figs. 13 14, 15, 16, 17 and 18 show these constraints for each compared solution (Tables 9, 10, 11, 12, 13, 14).

From these curves, it is possible to observe those constraints that are active or close to it and the inactive ones. Besides, the curves show the similarities of the solutions used for comparisons.

In these figures there are vertical lines dividing the x axes in two parts: the left part (including the point at the vertical line) corresponds to the displacement constraints, and the right one to the stresses constraints.

Table 14 Final design variables and objective function (weight—W) found for the 72-bar truss (TP6)

Variables	This study	SSGA (Lemonge et al. 2015)	MBA (Sadollah et al. 2012)	GAOS (Erbatur and Hasançebi 2000)	BGAwEIS (Talaslioglu 2009)
1	0.155878	0.153477	0.196000	0.155000	0.156000
2	0.565341	0.567966	0.563000	0.535000	0.555000
3	0.403595	0.389219	0.442000	0.480000	0.370000
4	0.567440	0.521187	0.602000	0.520000	0.510000
5	0.512861	0.548322	0.442000	0.460000	0.620000
6	0.507786	0.527722	0.442000	0.530000	0.530000
7	0.100000	0.100000	0.111000	0.120000	0.100000
8	0.100268	0.142985	0.111000	0.165000	0.100000
9	1.247423	1.502253	1.266000	1.155000	1.250000
10	0.508832	0.532541	0.563000	0.585000	0.523000
11	0.100061	0.100000	0.111000	0.100000	0.101000
12	0.100000	0.100000	0.111000	0.100000	0.105000
13	1.887894	1.714703	1.800000	1.755000	1.860000
14	0.516307	0.473252	0.602000	0.505000	0.513000
15	0.100000	0.100000	0.111000	0.105000	0.100000
16	0.100280	0.100000	0.111000	0.155000	0.100000
W	**379.655**	382.018	390.730	385.76	380.730
nfe	35000	30000	11600	20000	300000

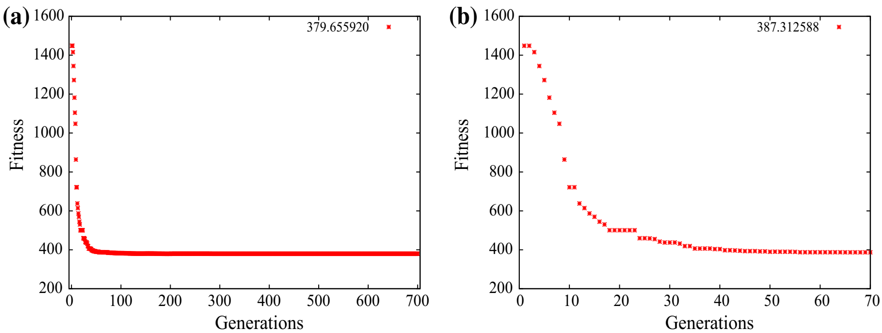


Fig. 12 Convergence for the 72-bar truss. **a** Using 700 generations (best = 379.655920) and **b** using 70 generations (best = 387.312588)

Furthermore, the curves presented in the plots inform the active constraints and the natures of them, displacements (left part) or normal stresses (right part). For example, observing the plots of Figs. 13 and 14, it is easy to note that two displacement constraints (2 and 4) and one stress constraint (13) are active (very close to “0”). From Fig. 15, two constraints (2 and 4) with respect to the

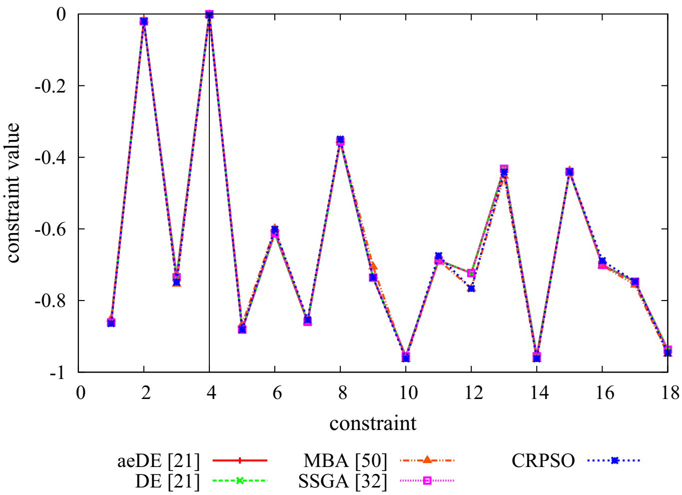


Fig. 13 Constraints for TP1—10-bar truss (discrete case). In this figure, and in Figs. 14, 15, 16, 17 and 18 the *vertical lines* (where it exists), divides the *x* axes in two parts: the *left part* (including the point at the vertical line) corresponds to the displacement constraints, and the right one to the stresses constraints

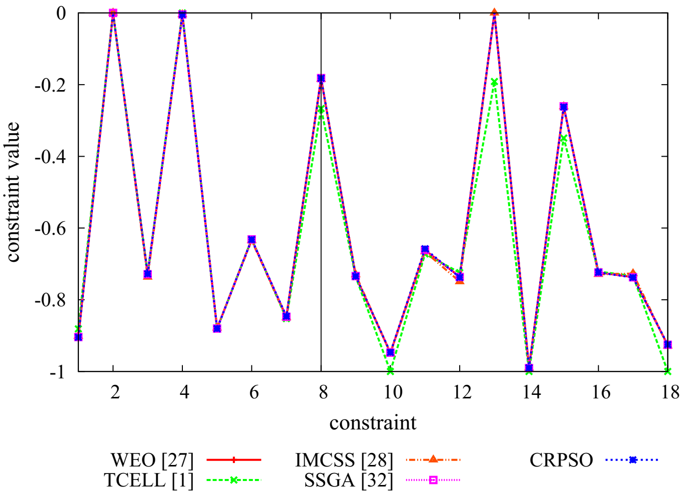


Fig. 14 Constraints for TP2—10-bar truss (continuous case)

displacements are active. From Fig. 16, there are five active or almost active constraints, and so on, when observing the other curves. Also, for each solution, it was calculated the sum of the absolute values of the constraints (SC) and, as expected, comparing the solutions, the solution that has the highest value for this sum will be the best among them. These plots are presented in Figs. 13, 14, 15, 16, 17 and 18.

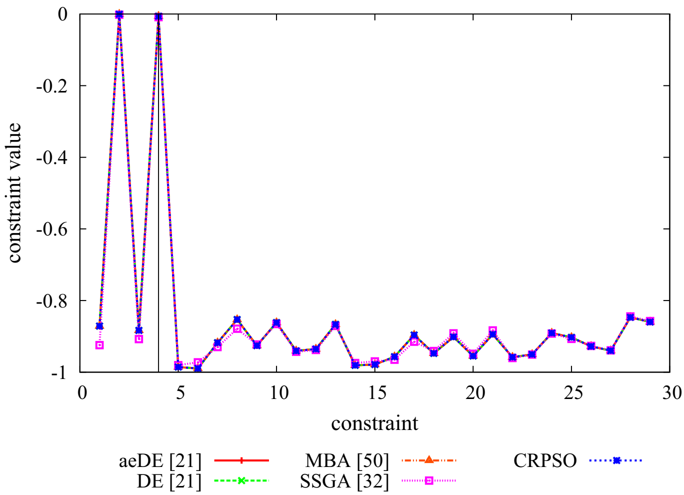


Fig. 15 Constraints for TP3—25-bar truss

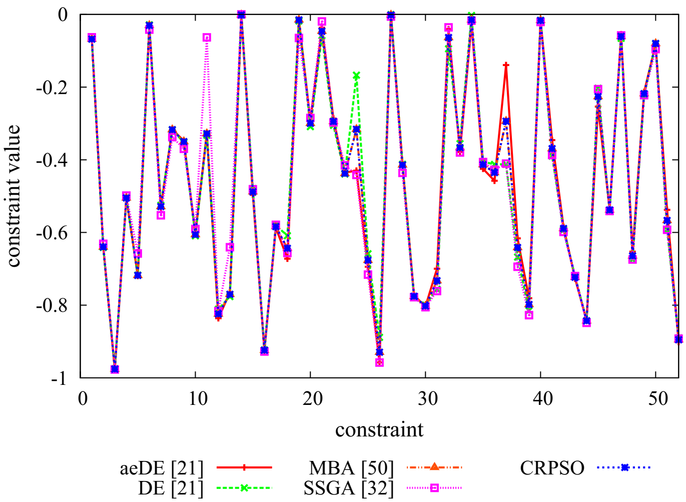


Fig. 16 Constraints for TP4—52-bar truss

It is very important to observe that the results presented in references Farshi and Alinia-Ziazi (2010) and Li et al. (2014), although slightly better than those, rigorously feasible, presented in this study and in the references Barbosa and Lemonge (2003) and Lemonge et al. (2015), are infeasible as shown in the curves displayed in Fig. 17. There are several constraints in these solutions greater than zero. Also, it is important to remark the difficulties to handle constraints either using

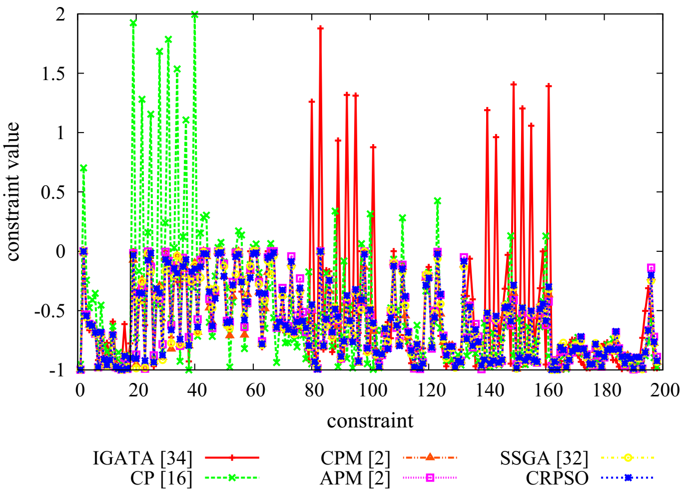


Fig. 17 Constraints for TP5—60-bar truss

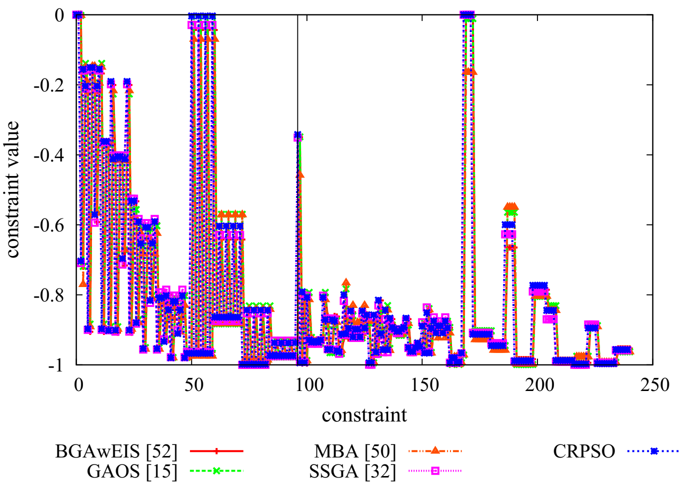


Fig. 18 Constraints for TP6—72-bar truss

a deterministic method such as in the reference Farshi and Alinia-Ziazi (2010) or an evolutionary algorithm such as in the reference Li et al. (2014).

A new metric is proposed in this paper where several solutions can be compared using the SC. It provides a way to verify a "distance" between the final design variables, the cross-sectional areas of the bars in the engineering optimization problems analyzed in this paper. One can observe from these curves that the solutions are quite similar since the curves are practically the same. Using this new

Table 15 Sum of the absolute values of the constraints (SC) for TP1, TP2 and TP3

TP1		TP2		TP3	
CRPSO	11.687819	CRPSO	10.892855	CRPSO	24.824018
SSGA (Lemonge et al. 2015)	11.642283	SSGA (Lemonge et al. 2015)	10.894458	SSGA (Lemonge et al. 2015)	24.910422
aeDE (Ho-Huu et al. 2016)	11.642283	IMCSS (Kaveh et al. 2015)	10.921155	aeDE (Ho-Huu et al. 2016)	24.824018
DE (Ho-Huu et al. 2016)	11.642283	WEO (Kaveh and Bakhshpoori 2016)	10.897381	DE (Ho-Huu et al. 2016)	24.824018
MBA (Sadollah et al. 2012)	11.689038	TCELL (Aragón et al. 2010)	11.367268	MBA (Sadollah et al. 2012)	24.824018

Table 16 Sum of the absolute values of the constraints (SC) for TP4, TP5 and TP6

TP4		TP5		TP6	
CRPSO	23.864739	CRPSO	121.165158	CRPSO	196.601211
SSGA (Lemonge et al. 2015)	23.922283	SSGA (Lemonge et al. 2015)	122.566531	SSGA (Lemonge et al. 2015)	197.119539
aeDE (Ho-Huu et al. 2016)	23.793616	APM (Barbosa and Lemonge 2003)	121.016799	MBA (Sadollah et al. 2012)	198.525051
DE (Ho-Huu et al. 2016)	23.838532	CPM (Barbosa and Lemonge 2003)	123.279729	GAOS (Erbatur and Hasançebi 2000)	196.639829
MBA (Sadollah et al. 2012)	23.864739	CP (Farshi and Alinia-Ziazi 2010)	139.130030	BGAwEIS (Talaslioglu 2009)	197.453209
		IGATA (Li et al. 2014)	128.784990		

metric, SC indicates how close are the solutions. Tables 15 and 16 show the values of SC for each test problem showing that the solutions are quite similar.

6 Concluding remarks and future work

This paper discusses a particle swarm algorithm (PSO) to solve constrained structural optimization problems. An adaptive penalty method (APM) and several variants of it are used. The PSO used in this work presents a change from the conventional algorithm. An operator called "craziness velocity" is adopted and the new algorithm is denoted CRPSO.

Firstly, seven APM variants are proposed and analyzed in order to discover which ones present the best performance. Subsequently, the three best variants and the original APM are used in the second analysis. It is important to note that the new variants proposed in this paper have not been analyzed in previous works.

Considering that this paper has the major focus on the engineering optimization problems, the main interest of a designer is to find the best solution sometimes neglecting other metrics. In this way, from the results presented in Table 7, the Variant 3 achieved the best performance and its use is recommended.

Take note that the best variant found in Lemonge et al. (2015) was not the same as obtained in this study. However, the robustness of these two variants is very similar, presenting very competitive results. A further important point is that the proposed algorithm performs better when the problems have continuous variables (Test Problems 5 and 6) and the algorithm proposed in Lemonge et al. (2015) performed better when the variables are discrete (Test Problem 1).

For future work the development of a multi-objective PSO to solve structural optimization problems in which, for example, the maximum displacement of the nodes is a further objective to be minimized in searching for the minimum weight of the structure, is intended, as is testing the new suite of functions available in the literature (Liang et al. 2013), and checking the performance of the variants highlighted in this paper. Finally, it is intended to apply APM to structural optimization problems considering frequency constraints and optimum solutions considering the best member groupings via cardinality constraints (Barbosa et al. 2008).

Acknowledgements The authors thank CNPq (305175/2013-0 and 305099/2014-0) FAPEMIG (Grants TEC PPM 528/11, TEC PPM 388/14 and APQ 00103-12) and CAPES for their support.

References

Aragón VS, Esquivel SC, Coello CAC (2010) A modified version of a t-cell algorithm for constrained optimization problems. Int J Numer Methods Eng 84(3):351–378

Barbosa HJ, Lemonge AC (2003) A new adaptive penalty scheme for genetic algorithms. Inf Sci 156(3):215–251

Barbosa HJC (1999) A coevolutionary genetic algorithm for constrained optimization. In: Proceedings of the congress on evolutionary computation (CEC), vol 3. IEEE, p 1611

Barbosa HJC, Lemonge ACC (2002) An adaptive penalty scheme in genetic algorithms for constrained optimiazation problems. In: GECCO 2002: proceedings of the genetic and evolutionary computation conference. Morgan Kaufmann Publishers, New York, pp 287–294

Barbosa HJC, Lemonge ACC (2008) An adaptive penalty method for genetic algorithms in constrained optimization problems. Front Evol Robot 34:596

Barbosa HJC, Lemonge ACC, Borges CCH (2008) A genetic algorithm encoding for cardinality constraints and automatic variable linking in structural optimization. Eng Struct 30(12):3708–3723

Barbosa HJC, Bernardino HS, Barreto AMS (2010) Using performance profiles to analyze the results of the 2006 CEC constrained optimization competition. In: Proceedings of the IEEE congress on evolutionary computation (CEC). IEEE, pp 1–8

Barbosa HJC, Lemonge ACC, Bernardino HS (2015) A critical review of adaptive penalty techniques in evolutionary computation. In: Evolutionary constrained optimization. Springer, pp 1–27

Bernardino HS, Barbosa HJC, Fonseca LG (2011) Surrogate-assisted clonal selection algorithms for expensive optimization problems. Evol Intell 4(2):81–97

Coello CAC (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput Methods Appl Mech Eng 191(11–12):1245–1287

Dobslaw F (2010) A parameter-tuning framework for metaheuristics based on design of experiments and artificial neural networks. Int J Comput Electr Autom Control Inf Eng 4(4):75–78

Dolan ED, Moré JJ (2002) Benchmarking optimization software with performance profiles. Math Program 91(2):201–213

Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, 1995. MHS'95, IEEE, pp 39–43

Elsayed SM, Sarker R, Mezura-Montes E et al (2013) Particle swarm optimizer for constrained optimization. In: Proceedings of the IEEE congress on evolutionary computation (CEC), IEEE, pp 2703–2711

Erbatur F, Hasançebi O, Tütüncür İlke, Kılıç H (2000) Optimal design of planar and space structures with genetic algorithms. Comput Struct 75(2):209–224

Farshi B, Alinia-Ziazi A (2010) Sizing optimization of truss structures by method of centers and force formulation. Int J Solids Struct 47(18):2508–2524

Gallet C Salaun M, Bouchet E (2005) An example of global structural optimisation with genetic algorithms in the aerospace field. In: VIII international conference on computational plasticity

Garg H (2016) A hybrid PSO-GA algorithm for constrained optimization problems. Appl Math Comput 274(1):292–305

Gellatly RA, Berke L (1971) Optimal structural design. Technical report, DTIC document

Hinterding R, Michalewicz Z (1998) Your brains and my beauty: parent matching for constrained optimisation. In: World congress on computational intelligence (WCCI), proceedings of the IEEE international conference on evolutionary computation (CEC). IEEE, pp 810–815

Ho-Huu V, Nguyen-Thoi T, Vo-Duy T , Nguyen-Trang T (2016) An adaptive elitist differential evolution for optimization of truss structures with discrete design variables. Comput Struct 165:59–75

Hu X, Eberhart RC, Shi Y (2003) Engineering optimization with particle swarm. In: Proceedings of the IEEE swarm intelligence symposium (SIS). IEEE, pp 53–57

Innocente MS, Afonso SMB, Sienz J, Davies HM (2015) Particle swarm algorithm with adaptive constraint handling and integrated surrogate model for the management of petroleum fields. Appl Soft Comput 34:463–484

Jordehi AR, Jasni J, Wahab NIA, Kadir M, et al (2013) Particle swarm optimisation applications in facts optimisation problem. In: Proceedings of the IEEE international power engineering and optimization conference (PEOCO). IEEE, pp 193–198

van Kampen AH, Strom C, Buydens LM (1996) Lethalization, penalty and repair functions for constraint handling in the genetic algorithm methodology. Chemom Intell Lab Syst 34(1):55–68

Kar R, Mandal D, Mondal S, Ghoshal SP (2012) Craziness based particle swarm optimization algorithm for fir band stop filter design. Swarm Evol Comput 7:58–64

Kaveh A, Mirzaei B, Jafarvand A (2015) An improved magnetic charged system search for optimization of truss structures with continuous and discrete variables. Appl Soft Comput 28:400–410

Kaveh A, Bakhshpoori T (2016) A new metaheuristic for continuous structural optimization: water evaporation optimization. Struct Multidiscip Optim 54(1):23–43

Kennedy J, Kennedy JF, Eberhart RC (2001) Swarm intelligence. Morgan Kaufmann Publishers, Burlington

Koziel S, Michalewicz Z (1998) A decoder-based evolutionary algorithm for constrained parameter optimization problems. In: Proceedings of the international conference on parallel problem solving from nature (PPSN). Springer, pp 231–240

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

Lemonge ACC, Barbosa HJC (2004) An adaptive penalty scheme for genetic algorithms in structural optimization. Int J Numer Methods Eng 59(5):703–736

Lemonge AC, Barbosa HJ, Bernardino HS (2015) Variants of an adaptive penalty scheme for steady-state genetic algorithms in engineering optimization. Eng Comput 32(8):2182–2215

Li D, Chen S, Huang H (2014) Improved genetic algorithm with two-level approximation for truss topology optimization. Struct Multidiscip Optim 49(5):795–814

Liang J, Runarsson TP, Mezura-Montes E, Clerc M, Suganthan P, Coello CC, Deb K (2006) Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. Technical report, nature inspired computation and applications laboratory

Liang J, Qu B, Suganthan P (2013) Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization. Technical report 201311, computational intelligence laboratory, Zhengzhou University, Zhengzhou China and technical report, Nanyang Technological University, Singapore

Liu Z, Hui Q (2012) A constraint-handling technique for particle swarm optimization. In: World automation congress (WAC), 2012. IEEE, pp 1–6

Mazhoud I, Hadj-Hamou K, Bigeon J, Joyeux P (2013) Particle swarm optimization for solving engineering problems: a new constraint-handling mechanism. Eng Appl Artif Intell 26(4):1263–1273

Mezura-Montes E, Coello CAC, Landa-Becerra R (2003) Engineering optimization using simple evolutionary algorithm. In: Proceedings of the IEEE international conference on tools with artificial intelligence (ICTAI). pp 149–156

Mezura-Montes E, Coello CAC (2011) Constraint-handling in nature-inspired numerical optimization: past, present and future. Swarm Evol Comput 1(4):173–194

Michalewicz Z (1995) A survey of constraint handling techniques in evolutionary computation methods. In: Proceedings of the 4th annual conference on evolutionary programming. MIT Press, pp 135–155

Michalewicz Z, Schoenauer M (1996) Evolutionary algorithms for constrained parameter optimization problems. Evol Comput 4(1):1–32

Monson CK, Seppi KD (2005) Linear equality constraints and homomorphous mappings in PSO. IEEE Congr Evol Comput 1:73–80

Orvosh D, Davis L (1994) Using a genetic algorithm to optimize problems with feasibility constraints. In: Proceedings of the IEEE conference on evolutionary computation (CEC), IEE world congress on computational intelligence. IEEE, pp 548–553

Parsopoulos KE, Vrahatis MN (2002) Particle swarm optimization method for constrained optimization problems. In: Proceedings of the Euro-international symposium on computational intelligence. IOS Press, pp. 214–220

Patnaik S, Hopkins D, Coroneos R (1996) Structural optimization with approximate sensitivities. Comput Struct 58(2):407–418

Rajeev S, Krishnamoorthy C (1992) Discrete optimization of structures using genetic algorithms. J Struct Eng 118(5):1233–1250

Rocha A, Fernandes E (2009) Self-adaptive penalties in the electromagnetism-like algorithm for constrained global optimization problems. In: Proceedings of the 8th world congress on structural and multidisciplinary optimization. pp 1–10

Runarsson TP, Yao X (2000) Stochastic ranking for constrained evolutionary optimization. IEEE Trans Evol Comput 4(3):284–294

Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2012) Mine blast algorithm for optimization of truss structures with discrete variables. Comput Struct 102:49–63

Silva EK, Barbosa HJC, Lemonge ACC (2011) An adaptive constraint handling technique for differential evolution with dynamic use of variants in engineering optimization. Optim Eng 12(1):31–54

Talaslioglu T (2009) A new genetic algorithm methodology for design optimization of truss structures: bipopulation-based genetic algorithm with enhanced interval search. Model Simul Eng 2009:6

Tang K, Li X, Suganthan P, Yang Z, Weise T (2010) Benchmark functions for the CEC 2010 special session and competition on large-scale global optimization. Technical report, nature inspired computation and applications laboratory

Vargas D, Lemonge A, Barbosa H, Bernardino HS (2015) Um algoritmo baseado em evolução diferencial para problemas de otimização estrutural multiobjetivo com restrições. Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería. (in portuguese)

Venkayya V (1971) Design of optimum structures. Comput Struct 1(1):265–309

Venter G, Haftka RT (2010) Constrained particle swarm optimization using a bi-objective formulation. Struct Multidiscip Optim 40(65):65–76

Wang Y, Cai Z (2009) A hybrid multi-swarm particle swarm optimization to solve constrained optimization problems. Front Comput Sci China 3(1):38–52

Wu SJ, Chow PT (1995) Steady-state genetic algorithms for discrete optimization of trusses. Comput Struct 56(6):979–991