Received 3 October 2011
Revised 5 March 2012
2 July 2012
Accepted 3 July 2012

Particle swarm optimization and identification of inelastic material parameters

M. Vaz Jr, E.L. Cardoso and J. Stahlschmidt
Department of Mechanical Engineering,
State University of Santa Catarina, Joinville, Brazil

Abstract

Purpose – Parameter identification is a technique which aims at determining material or other process parameters based on a combination of experimental and numerical techniques. In recent years, heuristic approaches, such as genetic algorithms (GAs), have been proposed as possible alternatives to classical identification procedures. The present work shows that particle swarm optimization (PSO), as an example of such methods, is also appropriate to identification of inelastic parameters. The paper aims to discuss these issues.

Design/methodology/approach – PSO is a class of swarm intelligence algorithms which attempts to reproduce the social behaviour of a generic population. In parameter identification, each individual particle is associated to hyper-coordinates in the search space, corresponding to a set of material parameters, upon which velocity operators with random components are applied, leading the particles to cluster together at convergence.

Findings – PSO has proved to be a viable alternative to identification of inelastic parameters owing to its robustness (achieving the global minimum with high tolerance for variations of the population size and control parameters), and, contrasting to GAs, higher convergence rate and small number of control variables.

Originality/value – PSO has been mostly applied to electrical and industrial engineering. This paper extends the field of application of the method to identification of inelastic material parameters.

Keywords Parameter identification, Particle swarm optimization, Optimization techniques, Genetic algorithms

Paper type Research paper

1. Introduction

The recent progress on robust numerical modelling of elastic-plastic problems at finite strains and development of efficient remeshing procedures have made possible the simulation of a wide range of metal forming operations. In spite of such advancements, the success of a simulation is crucially associated with the constitutive relation and corresponding material parameters used to material modelling. In the last few years, it has been observed a steady growth in application of identification techniques to obtaining constitutive parameters based upon optimization methods. The literature shows that such identification procedures generally use gradient-based optimization and soft computing strategies. For instance, a Quasi-Newton gradient-based method was utilised by Mahnken (2002) to determine the yield stress parameters of a

Engineering Computations: International Journal for Computer-Aided Engineering and Software
Vol. 30 No. 7, 2013
pp. 936-960

0264-4401
DOI 10.1108/EC-10-2011-0118

The first author gratefully acknowledges the support provided by CNPq (National Council for Scientific and Technological Development – Project 301991/2009-0).

Gurson-type damage model. Aiming at a similar damage model, Springmann and Kuna (2005) proposed an identification strategy using the Levenberg-Marquardt gradient-based technique. Ponthot and Kleinermann (2006) presented an assessment of a hybrid approach combining several gradient-based optimization strategies to determine hardening parameters associated with the classical von Mises elastic-plastic material model.

Most authors have agreed that the high nonlinearity and non-convexity exhibited by most such optimization problems add further difficulties when attempting application of classical gradient-based methodologies (Muñoz-Rojas *et al.* (2011) and references therein). In addition, even in convex problems, some gradient-based schemes require also initial estimates relatively close to the final parameters due to convergence limitations, as discussed in Arora (2004). Strategies conceived to identification of inelastic parameters using soft computing algorithms have been proposed in an attempt overcome convergence difficulties and possible existence of multiple local minima. Soft computing methods do not require computation of gradients and are mostly based on heuristic concepts. However, a disadvantage of this class of algorithms lies on the high computational cost due to the large number of operations required. Evolutionary (e.g. GAs), swarm intelligence (e.g. particle swarm optimization (PSO) and ant colony optimization) and neural networks algorithms (e.g. artificial neural networks (ANN)) are amongst the most referred soft computing strategies for general engineering problems. Nevertheless, application to identification of elastic-plastic parameters is very limited or even non-existent for some of the methods. Following this route, Abendroth and Kuna (2006) introduced an identification procedure based on ANN and Aguir *et al.* (2011) used the ANN technique as an alternative to the finite element calculations to evaluate the objective function within the GA. Chaparro *et al.* (2008) combined a GA and the Levenberg-Marquardt method to determine hardening parameters of anisotropic materials. Muñoz-Rojas *et al.* (2010, 2011) discussed application of GAs to parameter identification of a Gurson-type damage model.

The brief review highlights the increase in use of gradient-free algorithms to identification of inelastic parameters due to the aforementioned benefits when addressing this class of inverse problems. The literature also shows that, regarding soft computing algorithms, identification of inelastic material parameters are mainly based on GAs and ANN. Unlike GA and ANN optimization methods, it is interesting to notice that few works discuss application of PSO to general structural engineering problems. This work, therefore, addresses identification of inelastic parameters for a von Mises material as a gateway to investigation of the suitability and performance of a PSO scheme for this class of problems. In addition, an instigating discussion on the (dis)advantages of the method when compared to GAs is also presented. This paper is organised as follows: Section 2 briefly presents the finite element formulation. Section 3 introduces the optimization problem and details, in Sections 3.1 and 3.2, the PSO and GA strategies, respectively. Section 4 presents the numerical examples. The concluding remarks are in Section 5.

2. The direct problem: elasto-plasticity at finite strains

The elastic-plastic mechanical boundary value problem (direct problem) is solved using the finite element formulation proposed by de Souza Neto *et al.* (2008). The approximation uses an updated Lagrangean formulation and an implicit time

integration scheme for the linear momentum equation. The discretized equations are derived from application of the principle of the virtual work to the linear equilibrium equation:

$$\delta W = \int_{\Omega} \left[\hat{\boldsymbol{\sigma}}\left(\boldsymbol{\alpha}_n, \boldsymbol{F}_{n+1}\right) : \nabla_x^s \boldsymbol{\eta} - \boldsymbol{b}_{n+1} \cdot \boldsymbol{\eta} \right] \mathrm{d}v - \int_{\partial\Omega} \boldsymbol{t}_{n+1} \cdot \boldsymbol{\eta} \, \mathrm{d}a = 0, \tag{1}$$

where Ω and $\partial\Omega$ denote integrals over the domain and boundary, respectively, $\boldsymbol{\eta}$ represents the virtual displacements, and $\boldsymbol{t}_{n+1}$ and $\boldsymbol{b}_{n+1}$ are surface tractions and body forces prescribed at time step $n+1$. The incremental constitutive relation, $\hat{\boldsymbol{\sigma}}$, accounts for the constitutive material model and is associated to the internal variables, $\boldsymbol{\alpha}_n$, and deformation gradient, $\boldsymbol{F}_{n+1}$. Discretization of equation (1) yields a nonlinear equation for the global displacement vector, $\boldsymbol{U}_{n+1}$, at pseudo-time, t_{n+1}:

$$\boldsymbol{R}(\boldsymbol{U}_{n+1}) = \boldsymbol{f}_{n+1}^{\text{int}} - \boldsymbol{f}_{n+1}^{\text{ext}} = \boldsymbol{0} \;\text{ with }\; \begin{cases} \boldsymbol{f}^{\text{int}} = \int_{\Omega_h} [\boldsymbol{B}^g]^T \hat{\boldsymbol{\sigma}} \, \mathrm{d}v \\ \boldsymbol{f}^{\text{ext}} = \int_{\Omega_h} [\boldsymbol{N}^g]^T \boldsymbol{b} \, \mathrm{d}v + \int_{\partial\Omega_h} [\boldsymbol{N}^g]^T \boldsymbol{t} \, \mathrm{d}a, \end{cases} \tag{2}$$

where $\boldsymbol{R}$ is the residual of the iterative problem at pseudo-time $n+1$, $\boldsymbol{f}_{n+1}^{\text{int}}$ and $\boldsymbol{f}_{n+1}^{\text{ext}}$ are the internal and external forces, respectively, $\boldsymbol{B}^g$ is the discrete symmetric gradient operator (or strain-displacement matrix), and $\boldsymbol{N}^g$ is the global interpolation matrix. The standard Newton-Raphson iterative method is used to solve equation (2):

$$\boldsymbol{K}\left(\boldsymbol{U}_{n+1}^{(k)}\right)\left[\Delta \boldsymbol{U}_{n+1}^{(k)}\right] = -\boldsymbol{R}\left(\boldsymbol{U}_{n+1}^{(k)}\right) \quad \text{and} \quad \boldsymbol{U}_{n+1}^{(k+1)} = \boldsymbol{U}_{n+1}^{(k)} + \Delta \boldsymbol{U}_{n+1}^{(k)}, \tag{3}$$

in which $\boldsymbol{K}$ is the tangent stiffness and the superscript (k) indicates the Newton-Raphson iteration step.

3. Parameter identification and the optimization problem

Optimization methods for multidimensional problems are based on several principles (Arora, 2004; Goldberg and Sastry, 2011). The most common approaches in structural engineering make use of mathematical programming techniques and heuristic algorithms. Parameter identification is a class of inverse problems and consists of finding a set of material parameters which minimizes the difference between experimental measures and corresponding computed responses with respect to a given norm. The computed response is obtained by solving the direct problem, whereas the experimental response corresponds to measurement of selected variables of the experiment. The present identification procedure in based upon the nonlinear unconstrained optimization problem:

$$\begin{aligned} \textit{Minimize} \qquad & g_0(\boldsymbol{p}) \qquad\qquad \boldsymbol{p} \in R^n \\ & p_i^{\text{inf}} \le p_i \le p_i^{\text{sup}} \quad i = 1, \ldots, n, \end{aligned} \tag{4}$$

where $\boldsymbol{p} \in R^n$ is the design vector (a vector $\boldsymbol{p} = \left[p_1 \;\; p_2 \;\; \ldots \;\; p_i \;\; \ldots \;\; p_n\right]^T$ containing n design variables, i.e. the material parameters), $g_0(\boldsymbol{p})$ is the objective function (fitness) to be minimized, and p_i^{sup} and p_i^{inf} are the upper and lower bounds of the design variables, respectively. The objective function used in this work is based on the relative quadratic difference between experimental and computed responses evaluated over the deformation process:

$$g_0(\boldsymbol{p}) = \sqrt{\frac{1}{N}\sum_{q=1}^{N}\left(\frac{R_q^{FEM}(\boldsymbol{p}) - R_q^{Exp}}{R_q^{Exp}}\right)^2}, \tag{5}$$

in which R^{Exp} is the experimental measure, $R^{FEM}(\boldsymbol{p})$ is the corresponding numerical response computed using a set of design variables $\boldsymbol{p}$, and N is the number of experimental points.

3.1 Particle swarm optimization

PSO is an optimization technique based upon swarm intelligence and attempts to model the collective behaviour of moving particles within the hyperspace defined by design variables. The PSO was introduced by Eberhart and Kennedy (1995) and Kennedy and Eberhart (1995) and uses concepts of social behaviour of populations. Although PSO and GA require an initial population, both methods differ significantly with respect to their evolution. The GA population is characterised by chromosomes and genes of individuals and evolves using genetic operators. On the other hand, the initial population of most PSO implementations remains unaltered during the optimization process, to which velocity operators are applied causing the particles to move in the space of design variables.

Since Eberhart and Kennedy's (1995) original proposition, PSO has been successfully used in problems as different as job shop scheduling, vehicle routing and image processing, amongst many others, as extensively discussed by Poli *et al.* (2007) and Blum and Li (2008). However, application of PSO to inverse problems in structural engineering is a new endeavour. To the present date, just a few studies in these fields have been reported, such as the pioneering works on truss (Perez and Behdinan, 2007; Kaveh and Talatahari, 2011a) and pressure vessel design (Kaveh and Talatahari, 2011b). It is also important to mention that PSO algorithms have been successfully used to approach identification of general material parameters. Physical problems range from identification of visco-elastic material parameters in rock mass modelling (Feng *et al.*, 2006) to dynamic properties of composite materials (Hornig and Flowers, 2007); however, most such applications address identification of thermal parameters. For instance, Ardakani and Khodadad (2009) presented a discussion on identification of the thermal conductivity using the boundary element method. Identification of thermal parameters associated with a heat source was presented by Cortes *et al.* (2009). Tian *et al.* (2011) proposed application of a PSO scheme to temperature-dependent thermal conductivity in transient heat conduction problems.

In parameter identification problems, the particle coordinates in the hyperspace are the material parameters of the constitutive model. The search region is limited by boundaries imposed at each direction through upper and lower bounds. Therefore, the particles are allowed to move freely inside the hyper-volume so that, when convergence is reached, the entire population (or a high fraction of particles) flocks close to a single location – the converged values of the material parameters.

3.1.1 General description of the PSO algorithm

The PSO method aims at producing computational intelligence by mimicking the behaviour of a population through a combination of individual cognitive abilities and social interactions. The initial population is randomly defined in the search space

followed by computation of the fitness for each individual. The strategy consists in attributing velocities to each particle accounting for its inertia, personal history and effects of neighbouring individuals. The inertia component assumes that each particle would tend to follow along the previous direction. The particle history is associated with its best position, i.e. the location in the hyperspace which provides its best fitness. The social behaviour of the population gives rise to global best neighbouring individuals which contribute to a new direction of the particle.

In spite of the relatively short time since the original proposition, the literature shows several different variants of the method, most of which defining new neighbourhood conditions and rules of particle interactions. Schutte and Groenwold (2005), Poli *et al.* (2007), Blum and Li (2008) and Sedighizadeh and Masehian (2009) discuss general principles of many of such variants and present further details on different implementations of the method.

The first step consists of generating a random population and corresponding initial velocities:

$$\begin{aligned} \boldsymbol{p}^{(0)} &= \left\{ \boldsymbol{p}_1^{(0)} \;\; \boldsymbol{p}_2^{(0)} \;\; \ldots \;\; \boldsymbol{p}_m^{(0)} \;\; \ldots \;\; \boldsymbol{p}_{n_p}^{(0)} \right\} \text{ and} \\ \boldsymbol{v}^{(0)} &= \left\{ \boldsymbol{v}_1^{(0)} \;\; \boldsymbol{v}_2^{(0)} \;\; \ldots \;\; \boldsymbol{v}_m^{(0)} \;\; \ldots \;\; \boldsymbol{v}_{n_p}^{(0)} \right\}, \end{aligned} \tag{6}$$

so that, for a given particle:

$$\boldsymbol{p}_m^{(0)} = \left[p_1^{(0)} \;\; p_2^{(0)} \;\; \ldots \;\; p_i^{(0)} \;\; \ldots \;\; p_n^{(0)} \right]^T \quad \text{and} \quad \boldsymbol{v}_m^{(0)} = \left[v_1^{(0)} \;\; v_2^{(0)} \;\; \ldots \;\; v_i^{(0)} \;\; \ldots \;\; v_n^{(0)} \right]^T, \tag{7}$$

where p_i is the particle coordinate, v_i is the velocity component, n is the number of material parameters to be determined, and n_p is the number of particles or population size. The most common PSO operators compute the new velocity, $\boldsymbol{v}^{(k+1)}$, and location, $\boldsymbol{p}^{(k+1)}$, of the particles as:

$$\begin{aligned} \boldsymbol{v}^{(k+1)} &= \boldsymbol{v}_w^{(k)} + \boldsymbol{v}_{ib}^{(k)} + \boldsymbol{v}_{gb}^{(k)} \\ \boldsymbol{v}^{(k+1)} &= w\boldsymbol{v}^{(k)} + \boldsymbol{U}(0, \varphi_1) \otimes \left(\boldsymbol{p}_{ib}^{(k)} - \boldsymbol{p}^{(k)} \right) + \boldsymbol{U}(0, \varphi_2) \otimes \left(\boldsymbol{p}_{gb}^{(k)} - \boldsymbol{p}^{(k)} \right), \\ \boldsymbol{p}^{(k+1)} &= \boldsymbol{p}^{(k)} + \boldsymbol{v}^{(k+1)} \end{aligned} \tag{8}$$

in which the superscript (k) indicates time step, w is the inertia parameter, $\boldsymbol{p}_{ib}$ and $\boldsymbol{p}_{gb}$ are the individual and global best locations, respectively, and $\boldsymbol{U}(0, \varphi_1)$ and $\boldsymbol{U}(0, \varphi_2)$ represent vectors of random numbers in the interval $[0,\varphi_1]$ and $[0,\varphi_2]$, where φ_1 is the cognitive parameter representing the effect of self-knowledge and φ_2 is the social parameter associated with the collective effect of the population. The operation $\langle \cdot \rangle \otimes \langle \cdot \rangle$ indicates a component-wise multiplication.

It is interesting to note the components of equation (8): the "momentum" of the particle, $\boldsymbol{v}_w^{(k)} = w\boldsymbol{v}^{(k)}$, represents its tendency to continue moving along the same direction. The "cognitive" component, $\boldsymbol{v}_{ib}^{(k)} = \boldsymbol{U}(0, \varphi_1) \otimes (\boldsymbol{p}_{ib}^{(k)} - \boldsymbol{p}^{(k)})$, represents the tendency of the individual to return to its best location. The "social" component of the velocity, $\boldsymbol{v}_{gb}^{(k)} = \boldsymbol{U}(0, \varphi_2) \otimes (\boldsymbol{p}_{gb}^{(k)} - \boldsymbol{p}^{(k)})$, represents the tendency of the particle to move towards the position of the best individual of the population or a predefined neighbourhood. Figure 1 shows the method using the assumption of two material

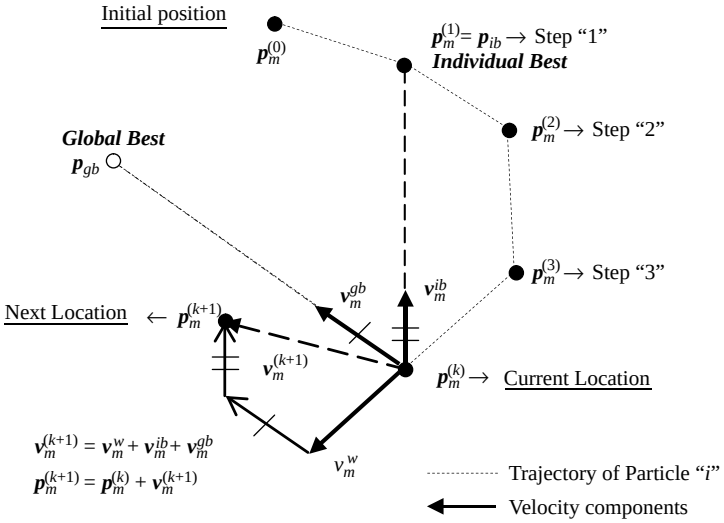

Figure 1.
Concept of PSO

parameters represented in a two-dimensional space, whereas "particle swarm optimization" shown below presents the PSO algorithm used in the parameter identification procedure.

Particle swarm optimization

(i) Set $k = 0$ and generate randomly the initial population $\boldsymbol{p}^{(0)}$ and corresponding velocities, $\boldsymbol{v}^{(0)}$:

$$\boldsymbol{p}^{(0)} = \left\{ \boldsymbol{p}_1^{(0)} \;\; \boldsymbol{p}_2^{(0)} \;\; \ldots \;\; \boldsymbol{p}_m^{(0)} \;\; \ldots \;\; \boldsymbol{p}_{n_p}^{(0)} \right\} \text{ and}$$
$$\boldsymbol{v}^{(0)} = \left\{ \boldsymbol{v}_1^{(0)} \;\; \boldsymbol{v}_2^{(0)} \;\; \ldots \;\; \boldsymbol{v}_m^{(0)} \;\; \ldots \;\; \boldsymbol{v}_{n_p}^{(0)} \right\}$$

in which n_p is the number of particles (individuals) and n is the number of design variables (material parameters):

$$\boldsymbol{p}_m^{(0)} = \left[p_1^{(0)} \;\; p_2^{(0)} \;\; \ldots \;\; p_i^{(0)} \;\; \ldots \;\; p_n^{(0)} \right]^T \text{ and}$$
$$\boldsymbol{v}_m^{(0)} = \left[v_1^{(0)} \;\; v_2^{(0)} \;\; \ldots \;\; v_i^{(0)} \;\; \ldots \;\; v_n^{(0)} \right]^T.$$

(ii) Evaluate individual and global best particles, $\boldsymbol{p}_{ib}^{(k)}$ and $\boldsymbol{p}_{gb}^{(k)}$:

$$\text{If } g_0\left(\boldsymbol{p}_m^{(k)}\right) < g_0\left(\boldsymbol{p}_{ib}^{(k)}\right) \Rightarrow \boldsymbol{p}_{ib}^{(k)} \leftarrow \boldsymbol{p}_m^{(k)}$$
$$\text{If } g_0\left(\boldsymbol{p}_m^{(k)}\right) < g_0\left(\boldsymbol{p}_{gb}^{(k)}\right) \Rightarrow \boldsymbol{p}_{gb}^{(k)} \leftarrow \boldsymbol{p}_m^{(k)}.$$

(iii) Compute new velocities $\boldsymbol{v}^{(k+1)}$:

$$\boldsymbol{v}^{(k+1)} = w\boldsymbol{v}^{(k)} + \boldsymbol{U}(0, \varphi_1) \otimes \left(\boldsymbol{p}_{ib}^{(k)} - \boldsymbol{p}^{(k)}\right) + \boldsymbol{U}(0, \varphi_2) \otimes \left(\boldsymbol{p}_{gb}^{(k)} - \boldsymbol{p}^{(k)}\right)$$

and verify against maximum limits of each design variable (material parameter):

$$v_i^{\max} = w_i\left(p_i^{\sup} - p_i^{\inf}\right) \quad \text{where} \quad w_i \in [0, 0.5]$$

$$\text{If} \left|v_i^{(k+1)}\right| > v_i^{\max} \Rightarrow \left|v_i^{(k+1)}\right| \leftarrow v_i^{\max}.$$

(iv) Compute new location of all particles $\boldsymbol{p}^{(k+1)}$ (design vectors):

$$\boldsymbol{p}^{(k+1)} = \boldsymbol{p}^{(k)} + \boldsymbol{v}^{(k+1)}$$

and verify boundary conditions $\left[p_i^{\sup}, p_i^{\inf}\right]$:

$$\text{If } p_i^{(k+1)} > p_i^{\sup} \Rightarrow p_i^{(k+1)} \leftarrow p_i^{\sup}$$

$$\text{If } p_i^{(k+1)} < p_i^{\inf} \Rightarrow p_i^{(k+1)} \leftarrow p_i^{\inf}$$

(v) Evaluate stopping criteria:

IF $\phi(\boldsymbol{p}) < TOL_{\phi}$ THEN

$\boldsymbol{p}^{\min} \leftarrow \boldsymbol{p}_{gb}^{(k+1)}$

EXIT

ELSE IF $k = k^{\max}$ THEN

EXIT

ELSE

$k \leftarrow k + 1$

GOTO (ii)

ENDIF

In addition to definition of new velocities and locations, as indicated in "particle swarm optimisation" shown above, the following issues have to be defined in PSO algorithms:

- velocity control;
- boundary conditions; and
- convergence or stopping criteria.

(i) Velocity control. The PSO method requires additional care in order to avoid excessive and unrealistic dispersion within the hyperspace of design variables. There is no unique set of rules to define such restrictions and the best approach is strongly problem dependent. In the present implementation, the velocity components are restricted to a fraction of the range, $(p_i^{\sup} - p_i^{\inf})$, for a given parameter, so that:

$$\text{If } \left|v_i^{(k+1)}\right| > v_i^{\max} \Rightarrow \left|v_i^{(k+1)}\right| \leftarrow v_i^{\max}, \quad \text{where} \quad v_i^{\max} = w_i\left(p_i^{\sup} - p_i^{\inf}\right) \tag{9}$$

is the maximum velocity component, $\left|v_i^{(k+1)}\right|$ is the absolute value of the velocity component, and w_i is the velocity restriction parameter, in which $0 < w_i \leq 0.5$. Clerc and Kennedy (2002) define a constriction parameter, χ:

$$\chi = \frac{2}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}} \quad \text{where} \quad \varphi = \varphi_1 + \varphi_2 > 4, \tag{10}$$

which multiplies the right hand side of equation (8) aiming at imposing a priori limitations for the particle velocity. However, numerical experiments have shown that such restriction, in the present parameter identification application, was insufficient to guarantee that particles at time step $(k + 1)$ be maintained inside of the search space without using boundary conditions.

(ii) Boundary conditions. Boundary conditions of a PSO optimization problem consist in re-defining the location of a particle $\boldsymbol{p}_m^{(k+1)}$ which eventually moves outside the pre-defined search limits, $p_i^{\inf} \leq p_i \leq p_i^{\sup}$. Despite using velocity control, if a particle is sufficiently close to the boundary of the search space, the random character of the velocity components may cause the particle to violate the pre-defined bounds. In the present simulation, as usually enforced in GA, a component-wise limit is imposed, i.e. the individual component is reset at the boundary, so that:

$$\begin{aligned} &\text{If } p_i^{(k+1)} > p_i^{\sup} \Rightarrow p_i^{(k+1)} \leftarrow p_i^{\sup} \\ &\text{If } p_i^{(k+1)} < p_i^{\inf} \Rightarrow p_i^{(k+1)} \leftarrow p_i^{\inf} \end{aligned}. \tag{11}$$

It is noteworthy that appropriate boundary conditions improves the converge rate and add a further random component to the search, thereby helping the algorithm to avoid local minima.

(iii) Convergence or stopping criteria. There is a great discussion on the best convergence or stopping criteria in gradient-free algorithms. The literature shows that most criteria adopted by GAs are based on the total number of generations or the difference between the fitness of the worst and best individuals. Therefore, similarly to GA, a normalised fitness convergence index, ϕ_g, based on the worst and best individuals (of n_s best particles) can also be defined as:

$$\phi_g^{(k)} = \frac{g_0\left(\boldsymbol{p}^{(k)}\right)^{worst} - g_0\left(\boldsymbol{p}^{(k)}\right)^{best}}{g_0\left(\boldsymbol{p}^{(0)}\right)^{worst} - g_0\left(\boldsymbol{p}^{(0)}\right)^{best}}. \tag{12}$$

Notwithstanding, the nature of PSO suggests convergence/stopping criteria based on a measure of dispersion of particles or corresponding velocities. After sufficiently large number incremental steps, particles reduce velocity and dispersion, flocking near the minimum. Therefore, a convergence criterion based upon particle dispersion, ϕ_d, can be established as:

$$\phi_d^{(k)} = \frac{\sum_{m=1}^{n_s} \left| \boldsymbol{p}_{sb}^{(k)} - \boldsymbol{p}_m^{(k)} \right|}{\sum_{m=1}^{n_s} \left| \boldsymbol{p}_{sb}^{(0)} - \boldsymbol{p}_m^{(0)} \right|}, \tag{13}$$

where $|< \cdot >|$ indicates the Euclidean norm, $\boldsymbol{p}_{sb}^{(k)}$ is the best particle and n_s is a fraction of the total population, defined in order to exclude non-physical individuals from computation of convergence indices.

Alternatively, the present implementation defines also a maximum number of steps/generations, $k^{\max}$, as a stopping criterion in order to further assess the evolution of the identification process. It is important to mention that equations (12) and (13) evaluate convergence only, without assessing the quality of the solution.

3.2 Genetic algorithms

It has been observed a widespread and increasing use of GAs to solve inverse problems owing to their capacity to find global minima without resorting to derivative computation. In recent years, there are many research works dealing with inverse problems and GAs, e.g. powder compaction (Kohei *et al.*, 2010), and identification of composite (Ahangar-Asr *et al.*, 2011), organic tissue (Khalil *et al.*, 2006), viscoelastic (Guo *et al.*, 2009), elastic-plastic (Chaparro *et al.*, 2008; Aguir *et al.*, 2011) and damage (Muñoz-Rojas *et al.*, 2010, 2011) parameters.

GAs are a subset of the broader class of evolutionary algorithms, in which each potential solution (individual) is represented by a vector of design variables (chromosome) and each design variable (gene) is represented by an information encoding system (e.g. binary encoding). As in the PSO method, an initial set of potential solutions (population) is randomly generated (equations (6) and (7)), so that a design variable (gene) is represented by:

$$p_i^{(0)} = (b_{n_b-1} \ \ldots \ b_r \ \ldots \ b_0), \tag{14}$$

where n_b is the number of bits b used to represent the design variable. Associated to each potential solution, there is a fitness value:

$$\boldsymbol{c}^{(0)} = \left\{ g_0\left(\boldsymbol{p}_1^{(0)}\right) \ g_0\left(\boldsymbol{p}_2^{(0)}\right) \ \ldots \ g_0\left(\boldsymbol{p}_m^{(0)}\right) \ \ldots \ g_0\left(\boldsymbol{p}_{n_p}^{(0)}\right) \right\}. \tag{15}$$

The genetic algorithm is summarised as follows and involves selection of a subset of the current population (mating pool), followed by one-point crossover, mutation, and generation of a new population.

Genetic algorithm

(i) Set $k = 0$ and generate randomly the initial population $\boldsymbol{p}^{(0)}$:

$$\boldsymbol{p}^{(0)} = \left\{ \boldsymbol{p}_1^{(0)} \ \boldsymbol{p}_2^{(0)} \ \ldots \ \boldsymbol{p}_m^{(0)} \ \ldots \ \boldsymbol{p}_{n_p}^{(0)} \right\}$$

in which n_p is the number of individuals and n is the number of design variables (material parameters):

$$\boldsymbol{p}_m^{(0)} = \left[p_1^{(0)} \ p_2^{(0)} \ \ldots \ p_i^{(0)} \ \ldots \ p_n^{(0)} \right]^T.$$

(ii) Evaluate individuals:

$$\boldsymbol{c}^{(k)} = \left\{ g_0\left(\boldsymbol{p}_1^{(k)}\right)\ g_0\left(\boldsymbol{p}_2^{(k)}\right)\ \ldots\ g_0\left(\boldsymbol{p}_m^{(k)}\right)\ \ldots\ g_0\left(\boldsymbol{p}_{n_p}^{(k)}\right) \right\}.$$

(iii) Select the mating pool $\boldsymbol{R}^{(k)}$ by binary tournament:

$$\boldsymbol{R}^{(k)} = \left\{ \boldsymbol{p}_1^{(k)} \boldsymbol{p}_2^{(k)}\ \ldots\ \boldsymbol{p}_m^{(k)}\ \ldots\ \boldsymbol{p}_{m_p}^{(k)} \right\}$$

in which m_p is the number of individuals in the mating pool.

(iv) Generate a new set of individuals $\boldsymbol{C}^{(k)}$ from $\boldsymbol{R}^{(k)}$ using one-point crossover:

$$\boldsymbol{C}^{(k)} = G(\boldsymbol{R}^k).$$

(v) Modify $\boldsymbol{C}^{(k)}$ using random mutation:

$$\boldsymbol{M}^{(k)} = M(\boldsymbol{C}^k)$$

Evaluate individuals from $\boldsymbol{M}^{(k)}$:

$$\boldsymbol{c}_m^{(k)} = \left\{ g_0\left(\boldsymbol{M}_1^{(k)}\right)\ g_0\left(\boldsymbol{M}_2^{(k)}\right)\ \ldots\ g_0\left(\boldsymbol{M}_m^{(k)}\right)\ \ldots\ g_0\left(\boldsymbol{M}_{n_p}^{(k)}\right) \right\}$$

(vi) Apply the replacement operator by sorting $\boldsymbol{M}^{(k)} \cup \boldsymbol{p}^{(k)}$ and selecting the new population. Full elitism is used by choosing the first n_p best individuals:

$$\boldsymbol{p}^{(k+1)} = R(\boldsymbol{M}^k \cup \boldsymbol{p}^{(k)}).$$

(vii) Evaluate stopping criteria:

IF $\phi(\boldsymbol{p}) < TOL_\phi$ THEN

$\boldsymbol{p}^{\min} \leftarrow \boldsymbol{p}^{(k+1)}$

EXIT

ELSE IF $k = k^{\max}$ THEN

EXIT

ELSE

$k \leftarrow k + 1$

GOTO (ii)

ENDIF

In this work, each design variable (gene) is encoded with a reflected binary code, also known as Gray code (Chakraborty and Janikow, 2003; Rowe *et al.*, 2004). This encoding is used due to its desirable property of having a unitary Hamming distance as opposed to the traditional binary representation.

(1) *Mating pool:* The mating pool, $\boldsymbol{R}$, is selected by binary tournament. In this approach, for each $i \in [1, m_p]$, two random individuals are selected from $\boldsymbol{p}$ and are compared against their fitness values. The fittest is then selected and stored in the mating pool. In this work, a mating pool of size $m_p = n_p/2$ is used.

(2) *Crossover operator:* The genetic operator known as crossover, $\mathbf{C} = G(\boldsymbol{R})$, was also used, which consists in generating new individuals by recombination of two random individuals (parents), $[b1_{(n\times n_b)-1} \ \ldots \ b1_0]^T$ and $[b2_{(n\times n_b)-1} \ \ldots \ b2_0]^T$, of the mating pool. This implementation uses a one-point crossover, where a random position $r_b \in [0, (n \times n_b) - 1]$ is selected and two individuals (offsprings) are generated by mixing the genetic information (bits) of the two parents around the crossover point, as:

$$\begin{aligned} \boldsymbol{c}_1 &= [b1_{(n\times n_b)-1} \ \ldots \ b1_{r_b} \oplus b2_{r_b-1} \ \ldots \ b2_0]^T \\ \boldsymbol{c}_2 &= [b2_{(n\times n_b)-1} \ \ldots \ b2_{r_b} \oplus b1_{r_b-1} \ \ldots \ b1_0]^T \end{aligned}, \tag{16}$$

where $\oplus$ indicates the crossover point.

(3) *Mutation operator:* Mutation, $\boldsymbol{M} = M(\mathbf{C})$, is a fundamental operator in GA methods (Arabas, 2009; Lin and Gen, 2009) and its use makes the method less prone to get trapped in local minima. This work uses homogeneous mutation, where a fixed percentage of bits, m_f, in the total population (mutation rate) is changed randomly. As the total number of bits in $\boldsymbol{p}$ is given by $n_p \times n \times n_b$, the number of bits changed in each iteration is $n_{bm} = m_f \times n_p \times n \times n_b$. Thus, for each $i \in [1, n_{bm}]$, it is selected a random individual, $\boldsymbol{c}_r$, of $\boldsymbol{C}$ and applied a logical not in a randomly selected bit.

(4) *Replacement operator:* The replacement operation, $R(\boldsymbol{M} \cup \boldsymbol{p})$, is also a very important aspect of the implementation of a GA. In this work, both $\boldsymbol{p}$ and $\boldsymbol{M}$ are sorted together, and the n_p best individuals are selected to compound the next population, $\boldsymbol{p}^{(k+1)}$, corresponding to a full elitism. Although full elitism can lead to a premature convergence to local minima in general optimization problems, it is used here to further compare the dispersion characteristics of both GA and PSO algorithms. Furthermore, this approach leads to a faster convergence when compared to partial elitism or no elitism at all. Full elitism has also another desirable property: as some individuals are kept throughout generations, a record of previously computed individuals can be maintained in a database, avoiding future evaluations of their fitness. This is especially advantageous in later generations.

4. Example of application and discussion

The PSO technique is applied to identification of hardening parameters of a carbon steel based upon tensile tests. A PSO/GA comparative study is also discussed. The cylindrical specimens were prepared according to the ASTM E 8M-01 standard. The load-displacement curve was measured using an extensometer with $\ell_0 = 50$ mm initial gauge length and a crosshead speed $v_c = 1.0$ mm/min. Measurements have been performed during the whole deformation process up to the final fractured state (triaxial stress states take place after necking). Six specimens were tested and the median load-elongation curve was used for the identification procedure (Figure 2(a)). The geometrical model used in the finite element approximation encompasses the extensometer region only and, due to

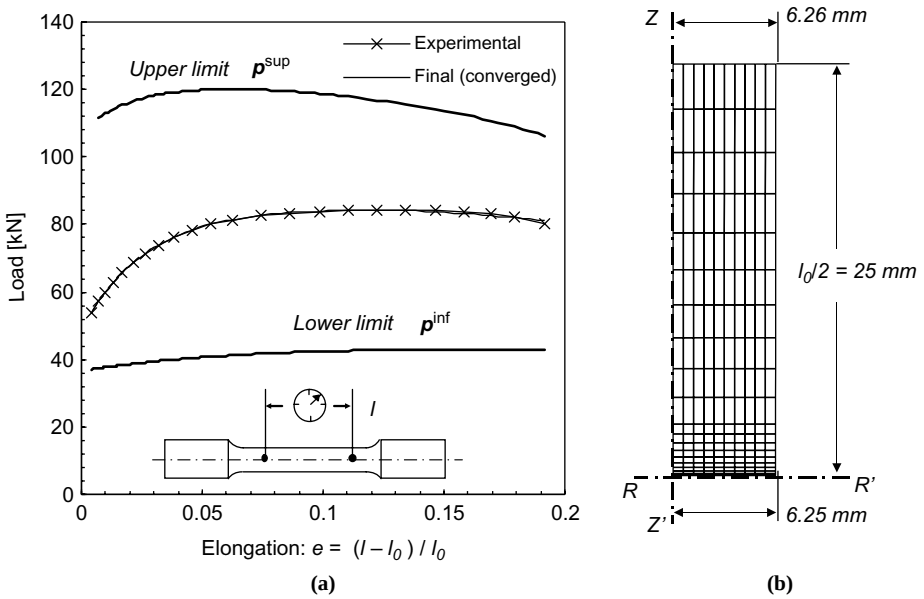

Figure 2.
Load-elongation curves and geometrical model

symmetry, only 1/4 of the specimen is modelled. The finite element mesh is shown in Figure 2(b) and uses 200 axisymmetric linear quadrilateral elements and 231 nodes, refined at the necking region. In the present example, the elastic parameters were assumed known in advance (Young modulus $E = 200$ GPa and Poisson's ratio $\nu = 0.3$). The elastic-plastic problem is complemented by the hardening curve which, in the present case, is based upon Voce's (1948) modified yield stress curve:

$$\sigma_Y = \sigma_0 + \zeta\varepsilon_p + (\sigma_\infty - \sigma_0)\left[1 - \exp\left(-\delta\varepsilon_p\right)\right], \tag{17}$$

in which σ_0, σ_∞, ζ and δ are the parameters to be determined. The literature shows that, for some carbon and alloy steels, equation (17) provides an accurate description of the yield stress curve associated with classical von Mises materials (Ponthot and Kleinermann, 2006). It is worthy to highlight that use of a classical nonlinear material to model the inelastic behaviour makes possible to focus the assessment on the identification problem itself, since computational issues of the constitutive modelling have already been well established.

The upper and lower limits, $p_i^{\inf}$ and $p_i^{\sup}$, of the parameters used in the optimization process are indicated in Table I. Figure 2(a) shows also the loading curves computed using the limiting parameters; however, such curves must be regarded as

	Parameter			
	σ_0 (MPa)	σ_∞ (MPa)	ζ (MPa)	δ
Lower limit, $p_i^{\inf}$	300.0	400.0	200.0	10.0
Upper limit, $p_i^{\sup}$	900.0	1,000.0	800.0	50.0
BFGS	421.9819	708.3987	592.5969	35.42862

Table I.
Lower and upper limits and converged parameters for the BFGS method

illustrations only since the PSO and GA methods are heuristic in nature and, upon generating the initial population, a combination of parameters with different values within the search space could yield more extreme curves.

Aiming at providing a trustworthy solution, the present test-case was also addressed using the well-known gradient-based BFGS optimization method (Arora, 2004). Such procedure enables, not only to determine the final fitness and reference solution (final parameters) for the present problem, but also to provide the PSO/GA methods verification grounds and convergence and robustness goals (Table I for the final BFGS parameters). In spite of the high success and convergence rate demonstrated by the BFGS method, numerical experiments based on the present identification problem suggest the existence of an envelope of initial parameter estimates (in the hyperspace of design variables) outside which convergence fails, i.e. convergence is ensured for initial estimates sufficiently close to the final parameters. The reader is referred to Arora (2004) for further issues on the BFGS scheme.

The investigation on the performance of the PSO technique comprises the following aspects:

- the general dynamics of the swarm behaviour;
- effect of the PSO control parameters upon global convergence;
- effect of population size; and
- comparisons with the GA optimization procedure.

In all cases, PSO convergence is assumed when the particle dispersion and fitness convergence indices reach $\phi_d^{conv} = 10^{-6}$ and $\phi_g^{conv} = 10^{-10}$, respectively.

4.1 The dynamics of the swarm behaviour

Unlike GAs, PSO defines a set of particles which retains their identities as individuals throughout the optimization process. This characteristic facilitates to study both the behaviour of the whole swarm and the trajectory of individual particles. Such analysis is highly recommended when addressing new neighbourhood search/restrictions, grouping, mobility, boundary conditions, velocity control and other performance-enhancing schemes. The relatively small number of parameters of the present problem helps visualization of the swarm dynamics using Cartesian coordinates. Although this example requires identification of four parameters, the swarm behaviour and particle trajectory are illustrated in the 3D Cartesian space for parameters σ_0, σ_∞ and ζ of equation (17). Very similar graphical representation is obtained when using parameter δ as one of the Cartesian coordinates.

Figure 3(a)-(d) shows the location of the particles and corresponding velocity vectors in the early stages of the identification process. Figure 3(a) shows the initial random location and velocity direction of the particles – the velocity vectors present different magnitudes and no preferential directions. The optimization progress shows that, early in the process, particles tend to cluster near the minimum, reducing dispersion of both direction and velocity magnitude, as shown in Figure 3(b)-(d). In this example, one cannot visually discern different particles after 30 time steps.

Convergence requires particles to consistently move towards the minimum as the optimization process advances. Analysis of the trajectory of particles reveals the particular dynamics of the problem being optimized. An erratic movement of the

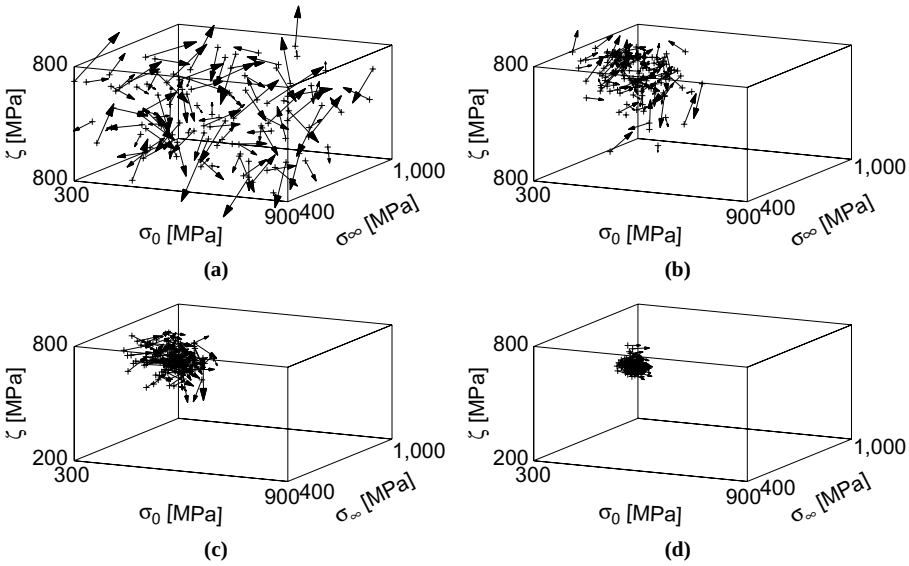

Figure 3.
Optimization process for a population of 120 particles and PSO control parameters $w = 0.5$ and $\varphi_1 = \varphi_2 = 1.0$

particle may indicate inappropriate control parameters, and/or require a more strict velocity control or even suffer from spurious effect of the neighbourhood search scheme. Figure 4 shows the trajectory of selected particles of the swarm reproduced in Figure 3. The depicted particles, denoted by "A", "B" and "C", present the largest

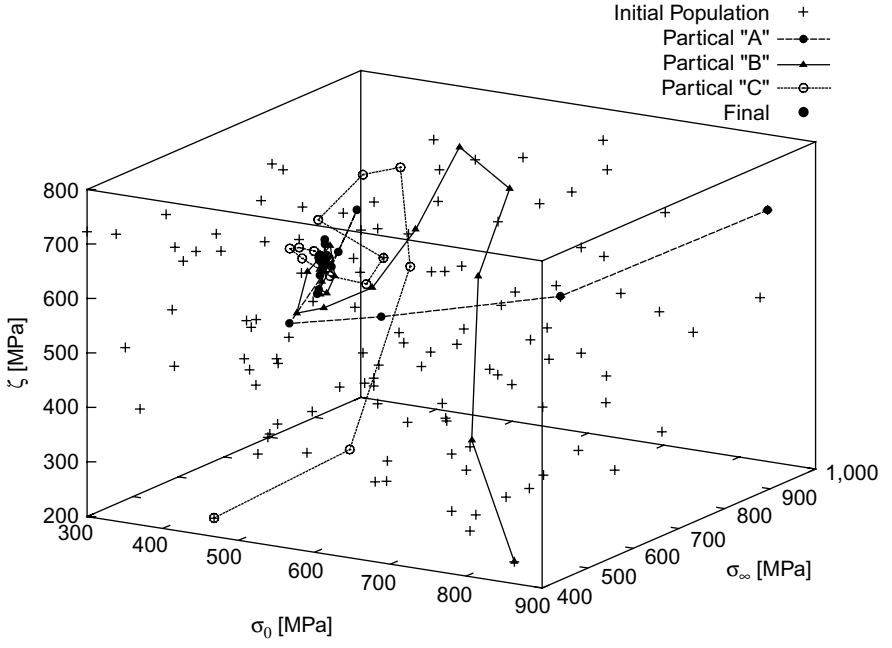

Figure 4.
Trajectories of particles "A", "B" and "C" initially located in different regions of the swarm

distance from the global minimum in the parameter hyperspace. The trajectories indicate that, early in the optimization process, particles move towards the minimum, with very few attempts to violate the search space.

4.2 Effect of PSO control parameters upon global convergence

The PSO control parameters for the original implementation of the algorithm are associated to inertia, cognitive and social components of the particle velocity. The weight attributed to the parameters defines the search scope of the algorithm and its ability to avoid spurious convergence and to prevent swarm instability. Unfortunately, there is no universal set of control weights which would guarantee success of the optimization process. In general, the inertia parameter affects the algorithm in its capacity to identify regions of potentially high quality solutions, i.e. large w facilitates global exploration and small values facilitates local search. On the other hand, the cognitive and social parameters exploit good solutions found in previous search steps, i.e. φ_1 and φ_2 define how quickly the algorithm can find a better solution within a restricted region.

Sedighizadeh and Masehian (2009) summarise large number of implementations and examples from several authors indicating the corresponding recommended values of the control parameters. Most authors suggest the inertia, cognitive and social parameters within the range $w \in [0.1, 1.0]$ and $\varphi_1 = \varphi_2 \in [0.5, 2.05]$, respectively. Therefore, the "best" set of control parameters is problem-dependent and must reflect an ideal balance between exploration and exploitation effects. This section summarises the investigation on the range of recommended values of w, φ_1 and φ_2 based upon the aforementioned considerations. Two sets of trials are described in the following paragraphs:

(1) The influence of w for a given set of φ_1 and φ_2 ($\varphi_1 = \varphi_2 = 1.0$).

(2) The effect of φ_1 and φ_2 for a given value of w ($w = 0.5$).

The example presents a population size $n_p = 120$ and $n_s = 114$ (95 per cent of the population).

(1) *The influence of* w:Figures 5 and 6 show evolution of the fitness and particle dispersion convergence indices, ϕ_g and ϕ_d, for inertia parameters $w = 0.1$, 0.3, 0.5, 0.7 and 0.9, and Table II presents the final converged parameters corresponding to the best particle (smallest fitness). The simulations show that small values of w decrease the capacity of the algorithm to perform an efficient search owing to poor exploration, leading the optimization process to a spurious convergence. On the other hand, particles tend to violate the limits of the search space for high values of w, causing the convergence rate to decreases substantially. Furthermore, it was also found that $w \geq 1$ may render the process unstable. Therefore, in the present case, the recommended range is $0.4 < w < 0.8$. It is noteworthy that the converged material parameters match those obtained using the gradient-based BFGS method. The load-elongation curve corresponding to the converged parameters is shown in Figure 2(a).

(2) *The influence of* φ_1 *and* φ_2*:* The cognitive and social weights define the learning capacity of the algorithm to find the minimum within a restricted region. The original studies by Kennedy and Eberhart (1995) proposed cognitive and social parameters $\varphi_1 = \varphi_2 = 2$ in an attempt to obtain an average value of 1 when multiplied by random numbers in the range of 0 and 1. Further general studies

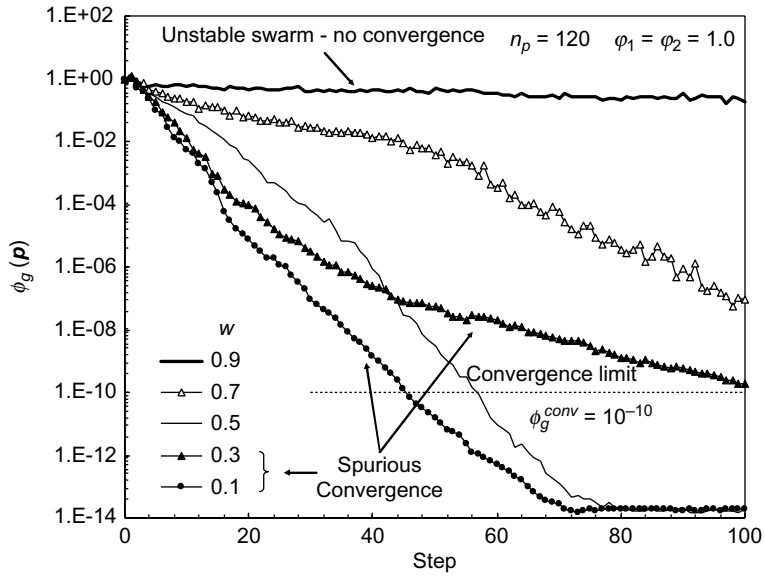

Figure 5.
Effect of the inertia parameter: fitness convergence index, ϕ_g

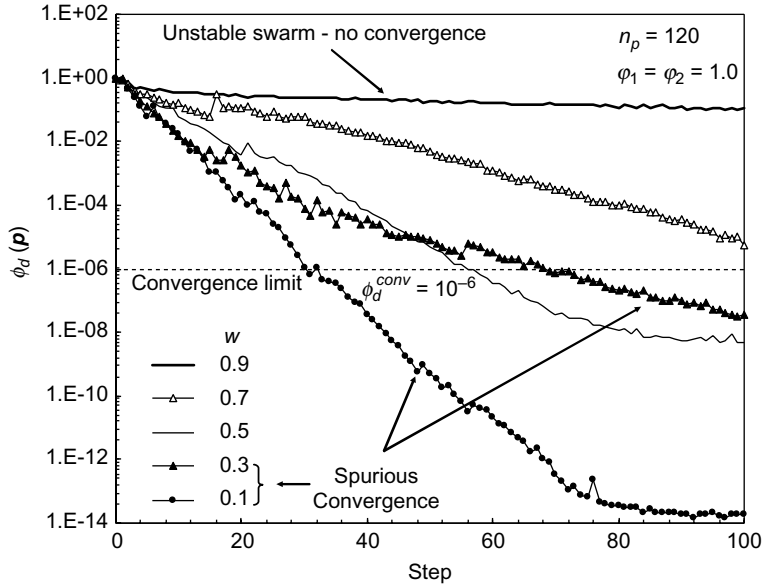

Figure 6.
Effect of the inertia parameter: particle dispersion index, ϕ_d

by Clerc and Kennedy (2002) introduced the constriction parameter, χ, defined in equation (10), requiring that $\varphi_1 + \varphi_2 > 4$. Most applications using this strategy suggest $\varphi_1 = \varphi_2 = 2.05$, so that the effective control parameters operating upon the velocity components are $w^{eq} = \chi w$, $\boldsymbol{U}\left(0, \varphi_1^{eq}\right) = \boldsymbol{U}(0, \chi\varphi_1)$

and $\boldsymbol{U}(0, \varphi_2^{eq}) = \boldsymbol{U}(0, \chi\varphi_2)$, in which $\chi\varphi_1 = \chi\varphi_2 \cong 1.4961$. In the present work, investigation on the cognitive and social weights was performed for $\varphi_1 = \varphi_2 =$0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8 and 2.0. The simulations have shown that larger values of φ_1 and φ_2 cause the algorithm to decrease its local search capacity, leading the convergence rate to decrease due to an erratic behaviour of the particles near target region, mutually affecting their learning capacity. On the other hand, the search can lead to spurious convergence for smaller values of φ_1 and φ_2 owing to poor exploitation. In such cases, the algorithm yields suboptimal parameters. The converged values (best particle) for the identified parameters are presented in Table III. Therefore, for the present example, the recommended cognitive and social parameters are $\varphi_1 = \varphi_2 \in [0.8, 1.8]$. It is interesting to mention that multiple runs of the PSO algorithm were performed being presented only typical results. In addition, the success rate for the recommended PSO control parameters in (i) and (ii) are 100 per cent, i.e. all runs obtained the reference material parameters.

4.3 Effect of the population size. Specific studies on the optimal population size are far more developed for GAs than PSO techniques. For instance, when addressing GA associated with a two-parameter test function, Arabas (2009) recommended two possible scenarios:

(1) smaller populations with full elitism and weaker selection pressure and generational replacement; and

(2) larger population sizes with stronger selection pressure and low elitism.

Table II.
Effect of the inertia: final parameters for $\varphi_1 = \varphi_2 = 1.0$

PSO weight	Parameter				
w	σ_0 (MPa)	σ_∞ (MPa)	ζ (MPa)	δ	Convergence ($\phi_g^{conv} = 10^{-10}$)
0.1	420.8047	705.2412	607.2365	36.31298	Spurious convergence
0.3	423.0307	710.8509	581.7737	34.73589	
0.5	421.9819	708.3987	592.5970	35.42863	Global minimum
0.7	421.9819	708.3987	592.5968	35.42862	
0.9	–	–	–	–	No convergence

Table III.
Effect of the cognitive and social parameters: final parameters for $w = 0.5$

PSO weight	Parameter				
$\varphi_1 = \varphi_2$	σ_0 (MPa)	σ_∞ (MPa)	ζ (MPa)	δ	Convergence ($\phi_g^{conv} = 10^{-10}$)
0.4	419.9570	704.0194	612.0206	36.78356	Spurious convergence
0.6	421.4926	707.5309	596.4004	35.70586	
0.8	421.9819	708.3985	592.5976	35.42866	
1.0	421.9819	708.3987	592.5970	35.42863	
1.2	421.9819	708.3987	592.5968	35.42862	Global minimum
1.4	421.9819	708.3987	592.5969	35.42863	
1.6	421.9819	708.3987	592.5968	35.42862	
1.8	421.9819	708.3987	592.5968	35.42862	
2.0	–	–	–	–	Unstable

Aiming at optimizing the number of generations, Gibbs *et al.* (2011) concluded based upon benchmark functions that GA would benefit with larger populations when a structured fitness function, and either epistatic interactions or at least one salient decision variable are present; otherwise a small population size is likely to produce the best results. Chen *et al.* (2012), based on an analytical approach of a multimodal problem, indicated that the assumption that larger GA populations increase diversity and, therefore, would increase rate of success, is not necessarily true. Contrasting with GA, besides PSO variants using resizable populations (Coelho and de Oliveira (2008) and references therein), only general aspects on optimal PSO population sizes have been investigated. One of the few works, based on classical benchmark problems, Shi and Eberhardt (1999) reported that the performance of the original PSO is not significantly sensitive to the population size. More recently, Hatanaka *et al.* (2009), in an attempt to evaluate early convergence to local optima, postulated that optimal population size is problem dependent.

Figures 7 and 8 show the influence of the population size evaluated using the fitness convergence index, ϕ_g, for the number of iteration steps and total number of fitness evaluations. The PSO control parameters are $w = 0.5$, $\varphi_1 = \varphi_2 = 1.0$ with convergence assessed using 95 per cent of the (best) particles. Table IV presents the converged parameters and corresponding iteration step and number of fitness evaluations. One could assume that ideal populations would present smaller sizes, high convergence rate and high rate of success (achieving the global minimum with high tolerance for variations of the control parameters). The test runs show that smaller populations may lead to spurious convergence owing to a poorer search capacity, as indicate results for populations of 50 and 100 particles. In addition, populations larger than 120 particles are able to converge to the global minimum with the PSO control parameters within the range discussed in Section 4.2. The experiments with populations of 120, 150, 200 and 250 particles indicate that, despite higher robustness, the diversity provided by larger

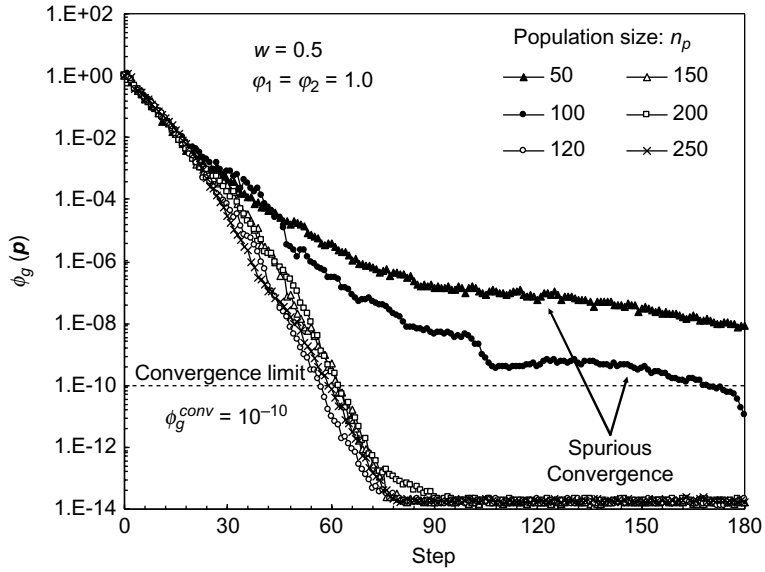

Figure 7.
Effect of the population size: evolution of the fitness convergence index, ϕ_g, with the iteration step

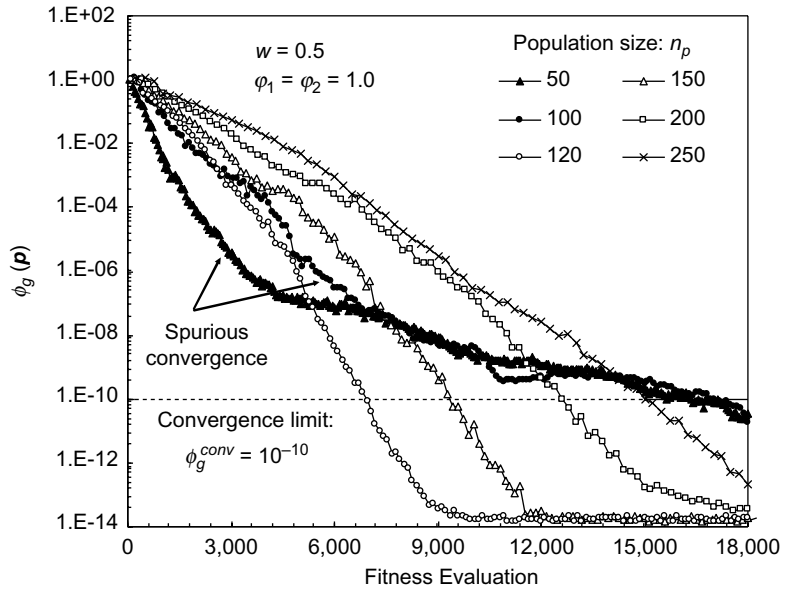

Figure 8.
Effect of the population size: evolution of the fitness convergence index, ϕ_g, with the number of fitness evaluations

Population	Parameter					Convergence ($\phi_g^{conv} = 10^{-10}$)	
n_p	σ_0 (MPa)	σ_∞ (MPa)	ζ (MPa)	δ	Step	Fitness evaluations	Minimum
50	425.4181	715.8650	558.2956	33.34682	326	16,350	Spurious convergence
100	422.0813	708.5798	591.7925	35.37520	170	9,400	
120	421.9819	708.3987	592.5969	35.42863	57	6,960	
150	421.9819	708.3987	592.5969	35.42862	62	9,450	Global minimum
200	421.9819	708.3987	592.5967	35.42861	62	12,600	
250	421.9819	708.3987	592.5969	35.42862	59	15,000	

Table IV.
Effect of the population size: converged parameters

population sizes does not accelerate convergence, i.e. convergence is reached in approximately the same iteration step, as summarised in Table IV (Figure 7), requiring, consequently, higher number of fitness evaluations (Figure 8).

4.4 Comparative example: GA and PSO optimization methods

Despite huge literature on developments and applications of heuristic algorithms to a wide range of optimization problems, use of such techniques to identification of elastic-plastic parameters is relatively recent. Chaparro *et al.* (2008) and Muñoz-Rojas *et al.* (2010, 2011) reported successful applications of GAs to this class of problems. However, the high number of fitness evaluations required to reach the global minimum apparently instigated the authors to use gradient-based optimization schemes when the process is sufficiently close to the optimum.

Comparative studies between GA and PSO algorithms have already been performed for general optimization problems. Elbeltagi *et al.* (2005) presented a comparison of

GAs, PSO, ant colony optimization, memetic algorithm and shuffled frog leaping algorithm using benchmark problems for continuous and discrete optimization. The authors concluded that the PSO method was generally better than the other algorithms in success rate and solution quality, while being second best in terms of processing time. A comparative study between GA and PSO in solving the human head tracking problem by Sulistijono and Kubota (2007) indicated a higher efficiency of the PSO method. Application of GA and PSO techniques by Toha and Tokhi (2010) to system identification of helicopter twin rotors evinced a better performance of PSO in accurately predicting the behaviour of the system. An exhaustive comparison between GA and PSO is beyond the scope of the present work; however, this example provides a first indication of the relative efficiency of the methods in identification of inelastic parameters. First, it is important to recall the different principles which rule both methods: PSO seeks the minimum through cooperation amongst particles whereas GA is based upon competition of individuals. The information sharing mechanisms are also different: the personal and global best particles give a one-way information direction to the swarm in classical implementations of PSO algorithms. In GA, the exchanging of chromosomes amongst individuals causes the population to move as a group towards the minimum. In spite of the random character of both techniques, all test runs demonstrate clear differences in evolution patterns.

The illustrative example makes use of the same population size, $n_p = 120$, with the following control parameters: GA uses 25 bits, a mutation rate of 5 per cent, one-point crossover with a probability of 80 per cent and full elitism. The PSO uses inertia, cognitive and social weights, $w = 0.5$ and $\varphi_1 = \varphi_2 = 1.0$, respectively. The dynamics of the GA require definition of less restrictive values of the dispersion and fitness convergence indices, $\phi_d^{GA} = 10^{-3}$ and $\phi_g^{GA} = 10^{-5}$.

Figure 9(a) and (b) shows the evolution of the convergence indices in the early stages of the identification process. The average dispersion around the best individual/particle is very similar in both methods, as shown in Figure 9(a). However,

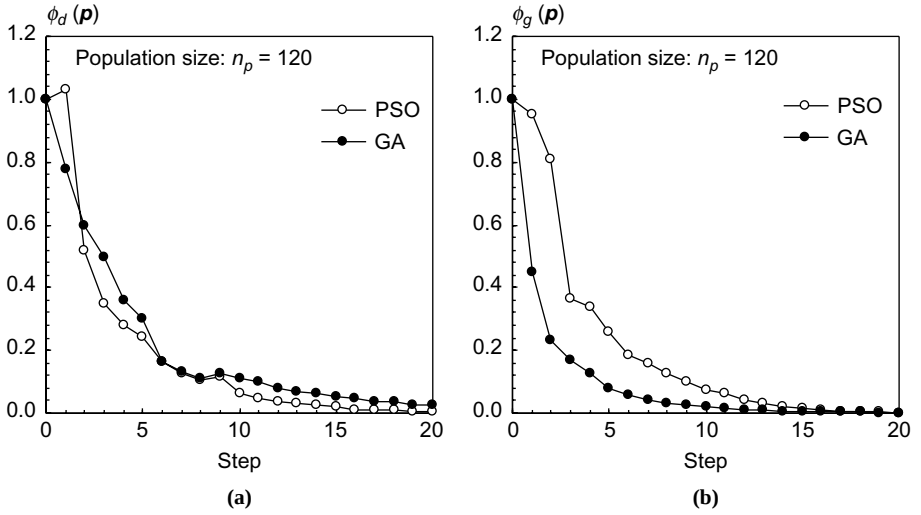

Figure 9.
Evolution of the (a) particle dispersion index, ϕ_d, and (b) fitness convergence index, ϕ_g, for PSO and GA at the early stages

the information sharing strategy of the PSO appears to be less efficient in the early stages, since the GA fitness index is generally smaller then its PSO counterpart, as exhibits Figure 9(b). The scenario changes dramatically when attempting to obtain the global minimum. Figure 10 demonstrates that the higher convergence rate of the GA in the early stages decreases substantially as the optimization evolves. Contrastingly, PSO attains a consistent rate of convergence up to $\phi_g 10^{-14}$. Convergence is achieved in solution steps (generations) $k = 52$ and $k = 57$ for GA and PSO methods, respectively, for a total run of 250 steps. Table V shows that the possibility to define stricter convergence conditions for PSO leads the identification to more accurate parameters – the PSO and GA fitness of the best particle presents variations in the 12th and fourth significant digits, respectively, (digit variation is italicised in Table V).

The authors acknowledge that a final conclusion on the comparative performance of GA and PSO techniques would require a more extensive investigation (using different mechanical tests, material modelling, materials, etc.); however, in the present example, the simulations markedly show that PSO presents higher convergence rate, being able to obtain material parameters with more rigorous convergence requirements. For the

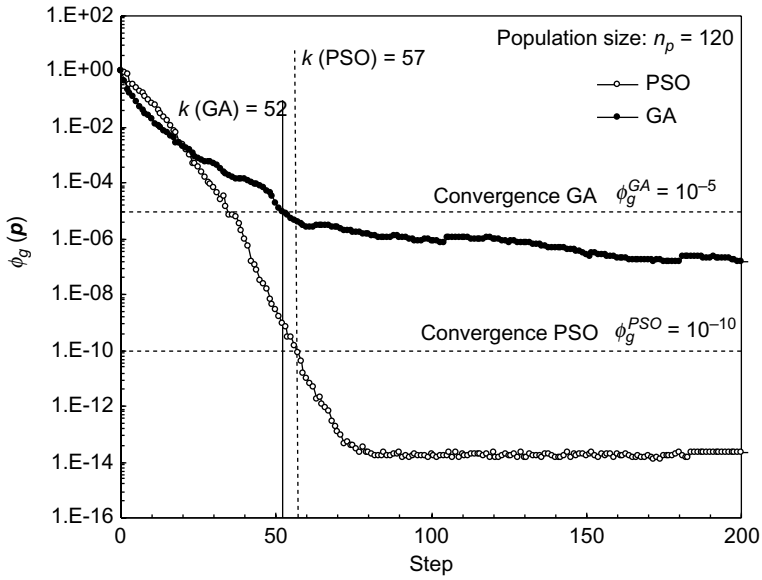

Figure 10.
Evolution of the fitness convergence index, ϕ_g, for PSO and GA

Method	Parameter σ_o (MPa)	σ_∞ (MPa)	ζ (MPa)	δ	Fitness $g_0(\boldsymbol{p}) \times 10^2$	Step/(fitness evaluations)
GA	422.0540	708.6170	591.4927	35.35742	0.266986136579	52 (6,360)
PSO	421.9819	708.3987	592.5969	35.42863	0.266693349076	57 (6,960)
BFGS	421.9819	708.3987	592.5969	35.42862	0.266693349074	14 (509)
GA	*422.0106*	*708.4855*	*592.2541*	*35.40449*	*0.266707712150*	*Final 250 (31,120)*
PSO	*421.9819*	*708.3987*	*592.5969*	*35.42862*	*0.266693349073*	

Table V.
Parameters at convergence: $\phi_g^{GA} = 10^{-5}$ and $\phi_g^{PSO} = 10^{-10}$

sake of verification, Table V also shows the converged parameters obtained using the BFGS gradient-based optimization method. It is noteworthy that PSO and BFGS objective functions at convergence differ in the 12th significant digit.

5. Concluding remarks

In recent years, parameter identification using optimization techniques has emerged as a preferred strategy to determine material constitutive parameters owing to their greater generality. Soft computing approaches, such as GAs and ANN, have been proposed as possible alternatives to identification procedures mainly due to their potential ability to avoid local minima and insensitivity to the order of magnitude of parameters. Easy parallelisation constitutes a great advantage of soft computing techniques, mitigating the requirement of larger number of fitness computation. Following recent suggestions on using such methods, the present work introduces application of PSO to identification of inelastic parameters.

As in other soft computing algorithms, the nature of the optimization problem, degree of nonlinearity and number of design variables dictate the behaviour of the PSO identification process and the recommended population size and other control parameters. For instance, use of more sophisticated material models or even a different yield stress curve may require larger (or allow smaller) population sizes. The numerical experiments indicate that, in the present example, the recommended PSO parameters are: population size $n_p = 120$, inertia weight in the range $0.4 < w < 0.8$ and cognitive and social control parameters $\varphi_1 = \varphi_2 \in [0.8, 1.8]$. Notwithstanding, the following general conclusions regarding the PSO control weights and population size can be used as guidelines to other identification problems:

- Smaller values of the inertia, w, or cognitive, φ_1, and social, φ_2, control parameters decrease the capacity of the PSO algorithm to perform an efficient search, leading the algorithm to spurious convergence.
- Excessively large values of w cause the particles to move in long and straight paths, which in turn, cause global instability and convergence difficulties.
- Excessively large values of φ_1 and φ_2 lead particles to move erratically around the target location causing instability of the swarm and also convergence difficulties.
- Smaller populations may also compromise the search capacity of the algorithm causing spurious convergence. Larger population sizes do not improve convergence rate, however robustness is increased with lesser chance of reaching a spurious convergence or getting trapped in local minima in non-convex problems.

In addition to the aforementioned observations, a brief comparative study between PSO and GA indicates that:

- Convergence rate for GA is higher in the early stages of the identification process. In the very first generations, the GA replacement operator quickly eliminates non-physical individuals, whereas the PSO particles retain their identities, so that unrealistic individuals require some steps to be brought inside the physically admissible region.

- In spite of such behaviour in the early steps, the PSO presents a consistent convergence rate and eventually outperforms the GA, being able to attain substantially more accurate results and stricter convergence requirements.

Therefore, the performance of the identification process indicates that PSO is competitive (especially when compared against GAs) and further studies using more complex material descriptions and multi-physics problems are highly recommended.

References

Abendroth, M. and Kuna, M. (2006), "Identification of ductile damage and fracture parameters from the small punch test using neural networks", *Engineering Fracture Mechanics*, Vol. 73, pp. 710-725.

Aguir, H., BelHadjSalah, H. and Hambli, R. (2011), "Parameter identification of an elasto-plastic behaviour using artificial neural networks-genetic algorithm method", *Materials and Design*, Vol. 32, pp. 48-53.

Ahangar-Asr, A., Faramarzi, A., Javadi, A.A. and Giustolisi, O. (2011), "Modelling mechanical behaviour of rubber concrete using evolutionary polynomial regression", *Engineering Computations*, Vol. 28, pp. 492-507.

Arabas, J. (2009), "Evolutionary method as a random tool for searching in R^n", *Computational Materials Science*, Vol. 45, pp. 21-26.

Ardakani, M.D. and Khodadad, M. (2009), "Identification of thermal conductivity and the shape of an inclusion using the boundary elements method and the particle swarm optimization algorithm", *Inverse Problems in Science and Engineering*, Vol. 17, pp. 855-870.

Arora, J.S. (2004), *Introduction to Optimum Design*, 2nd ed., Elsevier, Amsterdam.

Blum, C. and Li, X. (2008), "Swarm intelligence in optimisation", in Blum, C. and Merkle, D. (Eds), *Swarm Intelligence – Introduction and Applications*, Springer, Heidelberg, pp. 43-85.

Chakraborty, U.K. and Janikow, C. (2003), "An analysis of Gray versus binary encoding in genetic search", *Information Sciences*, Vol. 156, pp. 253-269.

Chaparro, B.M., Thuillier, S., Menezes, L.F., Manach, P.Y. and Fernandes, J.V. (2008), "Material parameters identification: gradient-based, genetic and hybrid optimization algorithms", *Computational Materials Science*, Vol. 44, pp. 339-346.

Chen, T., Tang, K., Chen, G. and Yao, X. (2012), "A large population size can be unhelpful in evolutionary algorithms", *Theoretical Computer Science*, Vol. 436, pp. 54-70.

Clerc, M. and Kennedy, J. (2002), "The particle swarm-explosion, stability, and convergence in a multidimensional complex space", *IEEE Transaction on Evolutionary Computation*, Vol. 6, pp. 58-73.

Coelho, A.L.V. and de Oliveira, D.G. (2008), "Dynamically tuning the population size in particle swarm optimisation", *Proceedings of the 2008 ACM Symposium on Applied Computing*, ACM Press, New York, NY, pp. 1782-1787.

Cortes, O., Urquiza, G. and Hernandez, J.A. (2009), "Inverse heat transfer using Levenberg-Marquardt and particle swarm optimization methods for heat source estimation", *Applied Mechanics and Materials*, Vol. 15, pp. 35-40.

de Souza Neto, E.A., Perić, D. and Owen, D.R.J. (2008), *Computational Methods for Plasticity: Theory and Applications*, Wiley, Chichester.

Eberhart, R. and Kennedy, J. (1995), "A new optimizer using particle swarm theory", *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, IEEE Press, Piscataway, NJ, pp. 39-43.

Elbeltagi, E., Hegazy, T. and Grierson, D. (2005), "Comparison among five evolutionary-based optimization algorithms", *Advanced Engineering Informatics*, Vol. 19, pp. 43-53.

Feng, X.-T., Chen, B.-R., Yang, C., Zhou, H. and Ding, X. (2006), "Identification of visco-elastic models for rocks using genetic programming coupled with the modified particle swarm optimization algorithm", *International Journal of Rock Mechanics and Mining Sciences*, Vol. 43, pp. 789-801.

Gibbs, M.S., Maier, H.R. and Dandy, G.C. (2011), "Relationship between problem characteristics and the optimal number of genetic algorithm generations", *Engineering Optimization*, Vol. 43, pp. 349-376.

Goldberg, D. and Sastry, K. (2011), *Genetic Algorithms: The Design of Innovation*, Springer, Heidelberg.

Guo, Y., Meng, G. and Li, H. (2009), "Parameter determination and response analysis of viscoelastic material", *Archive of Applied Mechanics*, Vol. 79, pp. 147-155.

Hatanaka, T., Korenaga, T., Kondo, N. and Uosaki, K. (2009), "Search performance improvement for PSO in high dimensional space", in Lazinica, A. (Ed.), *Particle Swarm Optimization*, In-Tech, Rijeka, pp. 249-260.

Hornig, K.H. and Flowers, G.T. (2007), "Performance of heuristic optimisation methods in the characterisation of the dynamic properties of sandwich composite materials", *International Journal of Acoustics and Vibrations*, Vol. 12, pp. 60-68.

Kaveh, A. and Talatahari, S. (2011a), "A general model for meta-heuristic algorithms using the concept of fields of forces", *Acta Mechanica*, Vol. 221, pp. 99-118.

Kaveh, A. and Talatahari, S. (2011b), "Hybrid charged system search and particle swarm optimization for engineering design problems", *Engineering Computations*, Vol. 28, pp. 423-440.

Kennedy, J. and Eberhart, R. (1995), "Particle swarm optimization", *Proceedings of the IEEE International Conference on Neural Networks*, IEEE Press, Piscataway, NJ, pp. 1942-1948.

Khalil, A.S., Bouma, B.E. and Mofrad, M.R.K. (2006), "A combined FEM/genetic algorithm for vascular soft tissue elasticity estimation", *Cardiovascular Engineering*, Vol. 6, pp. 93-102.

Lin, L. and Gen, M. (2009), "Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation", *Soft Computing*, Vol. 13, pp. 157-168.

Mahnken, R. (2002), "Theoretical, numerical and identification aspects of a new model class for ductile damage", *International Journal of Plasticity*, Vol. 18, pp. 801-831.

Muñoz-Rojas, P.A., Cardoso, E.L. and Vaz, M. Jr (2010), "Parameter identification of damage models using genetic algorithms", *Experimental Mechanics*, Vol. 50, pp. 627-634.

Muñoz-Rojas, P.A., da Cunda, L.A.B., Cardoso, E.L., Vaz, M. Jr and Creus, G.J. (2011), "A mixed optimization approach for parameter identification applied to the Gurson damage model", in Vaz, M. Jr, de Souza Neto, E.A. and Muñoz-Rojas, P.A. (Eds), *Advanced Computational Materials Modeling: From Classical to Multi-scale Techniques*, Wiley, Weinheim, pp. 165-204.

Perez, R.E. and Behdinan, K. (2007), "Particle swarm approach for structural design optimisation", *Computers & Structures*, Vol. 85, pp. 1579-1588.

Poli, R., Kennedy, J. and Blackwell, T. (2007), "Particle swarm optimization – an overview", *Swarm Intelligence*, Vol. 1, pp. 33-57.

Ponthot, J.-P. and Kleinermann, J.-P. (2006), "A cascade optimization methodology for automatic parameter identification and shape/process optimization in metal forming simulation", *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, pp. 5472-5508.

Rowe, J., Whitley, D., Barbulescu, L. and Watson, J.P. (2004), "Properties of Gray and binary representations", *Evolutionary Computation*, Vol. 12, pp. 47-76.

Schutte, J.F. and Groenwold, A.A. (2005), "A study of global optimization using particle swarms", *Journal of Global Optimization*, Vol. 31, pp. 93-108.

Sedighizadeh, D. and Masehian, E. (2009), "Particle swarm optimization methods, taxonomy and applications", *International Journal of Computer Theory and Engineering*, Vol. 1, pp. 486-502.

Shi, Y. and Eberhardt, C. (1999), "Empirical study of particle swarm optimization", *Proceedings of the IEEE Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, pp. 1945-1950.

Springmann, M. and Kuna, M. (2005), "Identification of material parameters of the Gurson-Tvergaard-Needleman model by combined experimental and numerical techniques", *Computational Materials Science*, Vol. 32, pp. 544-552.

Sulistijono, I.A. and Kubota, N. (2007), "Human head tracking based on particle swarm optimization and genetic algorithm", *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 11, pp. 681-687.

Tian, N., Sun, J., Xu, W. and Lai, C.-H. (2011), "Quantum-behaved particle swarm optimization with ring topology and its application in estimating temperature-dependent thermal conductivity", *Numerical Heat Transfer, Part B*, Vol. 60, pp. 73-95.

Toha, S.F. and Tokhi, M.O. (2010), "Parametric modelling application to a twin rotor system using recursive least squares, genetic, and swarm optimization techniques", *IMechE Journal of Aerospace Engineering*, Vol. 224, pp. 961-977.

Voce, E. (1948), "The relationship between stress and strain for homogeneous deformation", *Journal of Institute of Metals*, Vol. 74, pp. 537-562.

Further reading

Khoei, A.R., Keshavarz, S. and Khaloo, A.R. (2010), "The genetic algorithm approach for shape optimization of powder compaction processes considering contact friction and cap plasticity models", *Engineering Computations*, Vol. 27, pp. 322-353.

Corresponding author

M. Vaz Jr can be contacted at: M.Vaz@joinville.udesc.br