Particle Swarm Optimization for damage identification in beam-like structures

Authors: Alberto Barontini[a*], Maria Giovanna Masciotta[a], Luis Ramos[a], Paulo Amado-Mendes [b], Paulo Lourenço [a]

Affiliation:
[a] ISISE, University of Minho, Department of Civil Engineering, Campus de Azurém, Guimarães, 4800-058, Portugal
E-mail: albe.barontini@gmail.com
mg.masciotta@gmail.com
lramos@civil.uminho.pt
pbl@civil.uminho.pt

[b] ISISE, University of Coimbra, Department of Civil Engineering, R. Luis Reis dos Santos 290, 3030-790 Coimbra, Portugal
E-mail: pamendes@dec.uc.pt

*** Corresponding author Alberto Barontini.**

Abstract
The main objectives of Structural Health Monitoring (SHM) are the characterization and the assessment of the health condition of structural systems. Combined with appropriate Damage Identification (DI) strategies, SHM aims to provide reliable information about the localization and quantification of the structural damage by using an inverse formulation approach, with the damage parameters being estimated from parametric changes in dynamical properties. Mathematically, an inverse problem consists of the optimization of a function which represents the "distance" between the experimental and the numerically-simulated features of the system. Such process requires the development of a mock-up numerical model fairly representative of the system and iteratively updated until a response as close as possible to the experimental one is provided. The minimization of the difference between measured and predicted features' values is the objective function, whose global minimum corresponds to the best adjustment of the model variables. Metaheuristic represents a large class of global methods for optimization purposes able to outperform traditional methods in the following aspects: ease of implementation, time consumption, suitability for non-linear, black-box and high-dimensional problems. The present paper analyses, through a numerical experimentation approach, the suitability of one of the best-known metaheuristics, i.e. the Particle Swarm Optimization (PSO) algorithm, for DI of beam-like structures. Modal properties are used to define the objective

function and various algorithm instances are tested across different problem instances to assess robustness and influence of the algorithm parameters.

1. Introduction

Nowadays, developing cost-effective and automatic strategies for the maintenance of built environment is becoming essential, as many existing structures and infrastructures are close to the end of their service life (or over) and the new ones are growing in number, size and complexity. Furthermore, the costs of downtime and failure, the risk of injuries and life losses, as well as the repercussions that such accidents may have on a higher level are almost unaffordable.

Structural Health Monitoring (SHM) is an ongoing field of research whose main aim is the implementation of strategies for the assessment of the health condition of a structural system and the prompt identification of damage – when no information about its location is available – in order to avoid unexpected breakdowns and serious economic and societal losses. Damage Identification (DI) strategies can be categorized according to five main goals of increasing complexity (Farrar & Worden, 2007): (1) detection of existence, (2) localization, (3) classification of the type, (4) quantification of the extent and (5) prognosis of the remaining service life.

To achieve the higher goals, meaning at least up to the quantification of the damage extent, an inverse model updating problem formulation is usually adopted, which consists in the minimization of an objective function defined in terms of discrepancies between the features extracted by operational modal analysis and those computed using a numerical or analytical model.

Experience demonstrated that Particle Swarm Optimization (PSO) algorithms, as other metaheuristics, are suitable for the model updating as they do not need any knowledge of the function or of its derivatives. Furthermore, there is no requirement regarding the characteristics of the objective function itself, such as derivability or continuity, nor about the variables which can be continuous, discrete or mixed. Being population-based metaheuristic algorithms, PSO can explore in parallel more possible solutions in the same run and being also global methods, their performance does not depend on the initial population of solutions.

Despite the advantages listed above, three main shortcomings of PSO are worth mentioning. First, these methods are considered as sub-optimal. There is no guarantee that the achieved solution is the optimal one, as well as there is no guarantee of the convergence to the overall optimum value. Therefore, there is a risk of local optima trapping and premature convergence. Second, the canonical version of the algorithm shows a tendency to suffer an uncontained increase of the velocity during the process, also called explosion of the particles. Third, PSO algorithms, as the metaheuristics in general, have a peculiar problem related to the parameter setting. Each algorithm, in fact, requires the definition of several parameters, whose values can significantly affect the final performance.

The version of the PSO algorithm hereafter analysed relies on the inertia weight term, added to the canonical formulation to prevent the explosion of the particles Still, a specific analysis is required to overcome the other two drawbacks. In literature, many strategies have been formulated to face the premature convergence. In the DI field, most of the developed studies usually relied on multistage approaches to reduce the number of candidate locations for the damages (Nanda, Maity, & Maiti, 2014; Seyedpoor, 2012) and/or on improved version or hybridization of the canonical PSO (Kang, Li, & Xu, 2012; Kaveh, Javadi, & Maniat, 2014; Vakil Baghmisheh, Peimani, Sadeghi, Ettefagh, & Tabrizi, 2012). Indeed, the size of the problem and the balance between exploration and exploitation (two key stages of metaheuristic optimization algorithms) are likely to affect the performance in this regard. For the sake of clarity, it is stressed that exploration refers to searching across the space collecting information and providing a diversification of the possible solutions, whereas exploitation refers to intensifying the investigation on a restricted area close to the best achieved solutions. However, the balance between exploration and exploitation also depends on the parameter setting, which is commonly an underrated task in the definition of the algorithm instance to use, although its importance is well-known (Adenso-Díaz & Laguna, 2006).

Driven by the above considerations, the two main objectives of this paper are: (1) to test the PSO formulation developed by Shi and Eberhart (Shi & Eberhart, 1998), one of the most basic and common version, to demonstrate its suitability for damage identification in beam-like structures; (2) to show the influence of parameter setting and how to perform it.

Using a basic version of the algorithm is important not only to exclude that the poor performances observed in its application are due to an incorrect setting, but also to gain a deep knowledge of the ongoing process, focusing on the aspects that need an improvement of the algorithm. The parameter setting is herein achieved through the Design Of the Experiment (DOE), an approach for planning experiments that aims at assessing the influence of different factors on the result so as to find their best combination.

All the numerical simulations and the finite element models (FEM) are implemented in DIAMA (TNO, Delft, The Netherlands) using a Python script for the routine.

2. Overview on PSO algorithms

PSO is a name which encompasses a group of optimization algorithms developed starting from the first formulation by Kennedy and Eberhart (Kennedy & Eberhart, 1995) in 1995.

As inferable by the name, PSO algorithms draw inspiration from the social behavior of a swarm of animals, as flock of birds or school of fishes, generally addressed as particles.

Regardless the following improvements, in the common framework of the method, each particle is identified by its position and its velocity and keeps memory of the best position ever visited by itself and by the one that went nearest to the target among all the swarm (in the so called Global or gbest-PSO) or only among its neighborhood (in the so called Local or lbest-PSO). Based on this information, each particle decides the velocity and moves to a new position. According to biological inspiration, the solution achieved, at the t-th iteration, by the i-th particle of the p agents of the swarm, is called position, $X_{\mathrm{i}}^{\mathrm{t}}$, and it is defined by the coordinates in a s-dimensional space, where s is the number of variables x_{ij}^{t} composing the solution: $X_{\mathrm{i}}^{\mathrm{t}} = \{x_{i1}^{t}, x_{i2}^{t}, .., x_{is}^{t}\},\ i = 1,2, \dots p$. The change ratio of the position is called velocity.

A typical PSO algorithm works as follows: (1) the number of particles is defined, positions and initial velocities are initialized, (2) the distance between the actual position and the target (objective function) for each particle is evaluated, (3) the personal best position and the best position ever reached by a member of the swarm are memorized or updated, (4) the velocity of each particle is updated and, finally, (5) each position is updated. The process is repeated until the termination criteria are met.

In the version of the algorithm hereafter analysed, the position is updated according to the following expression:

$$x_{ij}^{t+1} = x_{ij}^{t} + v_{ij}^{t+1} \qquad (1)$$

where v_{ij}^{t+1} is the velocity, x_{ij}^{t+1} is the position at the iteration $t + 1$ and x_{ij}^{t} the position at the iteration t. The particle velocity is computed by:

$$v_{ij}^{t+1} = wv_{ij}^{t} + C_1 rand_1 \left(Pbest_{ij} - x_{ij}^{t}\right) + C_2 rand_2 \left(Gbest_j - x_{ij}^{t}\right) \qquad (2)$$

where C_1 and C_2 are positive weighting coefficients called learning factors, used to balance the influence of individual and social experience; $rand_1$ and $rand_2$ are two random numbers varying in between zero and one; vector $Pbest_{ij},\ i = 1,2, \dots p,\ j = 1,2, \dots s$, is the best position ever reached by the i-th agent and vector $Gbest_j,\ j = 1,2, \dots s$ is the best position ever reached by the flock, namely the position of the agent nearest to the target since the beginning of the process.

The first term of the equation (2) represents the direction of the particle in the previous step (its inertia), instead the second and the third terms represent the individual learning and the collective interaction, respectively.

The inertia weight w was introduced by Shi and Eberhart (Shi & Eberhart, 1998) to control the velocity, balancing exploration and exploitation. An adaptive term is suggested for w, since iteratively variable values initially push on global search and then switch for a local search. An extensively used formulation is the following:

$$w = w_{max} - iter\frac{w_{max}-w_{min}}{iter_{max}} \quad (3)$$

where $iter$ indicates the current iteration, w_{max} is the initial weight, w_{min} is the final weight and $iter_{max}$ stands for maximum iteration number.

A further precaution to prevent the explosion of the particles consists of the limitation of their velocity range.

$$v_{min} \leq v_i \leq v_{max} \quad (4)$$

3. Design of the experiments

Damage Identification is a constrained black-box optimization problem, since the decision variables correspond to the damage extent in each location, with values varying between 0 (completely collapsed) and 1 (no damage) and the objective function is not analytically known thus a FEM simulation is required to estimate it. The design variables express the damage ratio in terms of reduction of the Young modulus E. To avoid numerical problems, the lower bound of E is limited to 0.1 and an identification precision of 1% is adopted, which means that the algorithm can distinguish between damage extents that differs by 1%. Thus, from 10% to 100%, each element can assume 91 different possible values of the damage extent and the problem size is 91^s, where s is the number of candidate locations.

In the present study, a beam discretized in 20 elements is used to test the proposed PSO algorithm. Each element is a candidate location, therefore the problem size results $PS = 91^{20} \approx 1.5 \cdot 10^{39}$. The experiment is designed according to a two-levels factorial approach (see Table 1). There are four analysed parameters: (1) coefficient C_1, (2) coefficient C_2, (3) final weight w_{min} and (4) population size p.

The first three factors are the variables of the updating formula. Although the most common setting uses $C_1 = C_2 = 2$ and $w_{min} = 0.4$ (Kang et al., 2012; Nanda et al., 2014; Seyedpoor, 2012), in the present work, the two constants C_1 and C_2 are tested at 1 and 3, in accordance with the common advice of using $C_1 + C_2 < 4$ to avoid particles explosion, derived from (Clerc & Kennedy, 2002). Considering a constant value of $w_{max} = 0.9$ in the experiment w_{min} assumes values of 0.9 and 0.4, meaning that two strategies are compared, one with constant inertia weight equal to 0.9 and the other with dynamic inertia weight decreasing from 0.9 to 0.4. The fourth parameter p is an index of the coverage of the problem space (PS). For values of $p \ll PS$ the random initialization of the particles ensures that each of them is a different initial attempt and no clusters around specific points of the problem space exist. Thus, the bigger the population p, the better the exploration, but this also reduces the speed of each iteration.

Table 1: summary of the parameters in the 2-levels factorial design.

Number	C_1	C_2	w_{min}	p
1	Low	Low	Low	Low
2	Low	Low	Low	High
3	Low	Low	High	Low
4	Low	Low	High	High
5	Low	High	Low	Low
6	Low	High	Low	High
7	Low	High	High	Low
8	Low	High	High	High
9	High	Low	Low	Low
10	High	Low	Low	High
11	High	Low	High	Low
12	High	Low	High	High
13	High	High	Low	Low
14	High	High	Low	High
15	High	High	High	Low
16	High	High	High	High

The study intends to assess how much such a reduction is and whether it compensates for the growth of computational cost. To this end, a number of particles p equal at least to the number of the design variables s would be required, but a value threefold greater than the minimum is usually suggested (Gerist & Maheri, 2016). In this experiment, both levels are analysed: s and $3 \cdot s$.

4. Application and discussion of the results

The case study used to set the parameters of the proposed PSO algorithm is a 10 meters long clamped-clamped steel beam with a 0.2x0.45 m^2 cross section, discretized in 20 elements (Figure 1). The adopted objective function equals:

$$F = \sum_{i=1}^{nm} \frac{(f_{e,i} - f_{n,i})^2}{f_{e,i}^2} + \sum_{i=1}^{nm} \sum_{j=1}^{nn} \left(|\varphi_{e,ji}| - |\varphi_{n.ji}| \right)^2 \quad (5)$$

where f_i and φ_{ji} are the i-th natural frequency and modal coordinates, respectively, and the subscripts e and n refer to numerical and experimental respectively. To simulate the reduced information usually available in real situations, only a few lower modes are used ($nm = 5$) and not all the DOFs of the numerical model are employed to extract the modal coordinates ($nn = 6$). The considered nodes are circled in red in Figure 1. The structural damage is numerically simulated by halving the value of the Young modulus of the 11th element so as to reproduce a damage scenario in the midpoint.

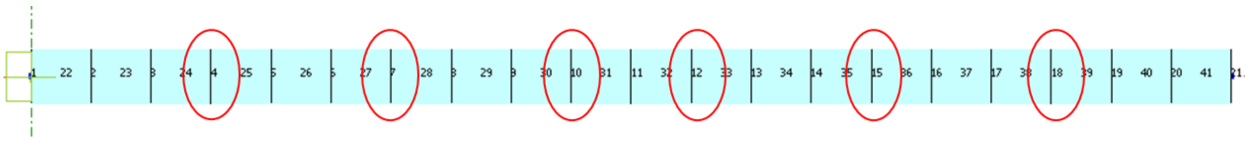

Figure 1: Case-study steel beam (the red circles indicate the DOFs considered in the optimization problem).

Throughout the experiments, the initial position and velocity of the particles are randomly generated. A random effect is also present at any iteration according to the formula of the velocity in Eq. (2). For any combination of factors, ten repetitions are carried out. No limitation to the velocity is introduced and two termination criteria are assumed: (1) a maximum number of iterations equal to 100 and (2) a value of the objective function equal to 0. In what concerns the latter criterion, allowing a margin through the definition of a threshold value is usually preferable, as setting for the objective function equal to 0 means looking for the exact solution rather than an optimal one. Still, the threshold value depends on the features used to define the function as well as on its formulation, thus setting a proper value should require a numerical experiment.

The performance of each repetition is assessed in terms of success (1=success, 0=failure) and number of FEM simulations. The latter is an index of the computational cost and corresponds to the number of particles multiplied by the number of iterations.

The set parameters are used as dependent variables in a univariate analysis of variance (ANOVA), analysing the influence of all the 4 independent factors on the algorithm performance. The null hypothesis H_0 corresponds to the case in which the coefficients C_1, C_2, w_{min} and the population size p do not affect the success of the PSO algorithm. Such hypothesis is rejected at a significance level of 5%, which means that if the $p - value$ calculated based on the observance value of the test statistic is less than the significance level the hypothesis is rejected. In this case, at least one of the factors or their interaction affects the performance.

In Table 2 the average results of the ten runs for each combination of factors are reported. The first four rows show, respectively, the success ratio over the ten runs, the average fitness in the ten runs, the average number of iterations required to achieve the solution (in case of failure, the process is repeated for 101 iterations, considering the initialization as first iteration), and the number of operations, viz. the number of FEM analysis. The other rows show the average damage extent identified by the algorithm for each element.

The results are analysed according to the ANOVA. From the test, it is possible to infer that the influence of p, C_1 and C_2 results statistically relevant. Moreover, a relevant correlation also exists between C_2 and w_{min} exists and between C_2 and N.

Table 2: PSO performance indicators and average values of the ten runs for each combination of the factorial design.

Test	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Succ. Ratio	0.8	0.9	0.6	0.7	0.3	0.8	0.5	0.8	0.9	1	0.9	1	0.4	0.8	0.7	1
Fit.	2.2E-03	5.1E-04	3.2E-03	1.1E-03	5.9E-03	1.0E-03	3.1E-03	2.0E-03	4.7E-04	0.0E+00	3.8E-04	0.0E+00	5.1E-03	1.0E-03	2.6E-03	0.0E+00
N° It	39.8	23.8	55.2	43.8	80.8	31.4	60.7	30	40.5	22.1	42	26.4	73	34.5	51.2	17.9
N° op	796	1428	1104	2628	1616	1884	1214	1800	810	1326	840	1584	1460	2070	1024	1074
El. 1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 2	1.00	1.00	1.00	0.98	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00
El. 3	1.00	1.00	0.99	0.98	1.00	0.99	1.00	1.00	1.00	1.00	0.99	1.00	0.98	1.00	0.99	1.00
El. 4	0.99	1.00	0.99	1.00	1.00	1.00	0.99	0.99	1.00	1.00	1.00	1.00	0.99	0.99	1.00	1.00
El. 5	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 6	0.99	1.00	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00	1.00
El. 7	1.00	1.00	1.00	0.99	1.00	1.00	0.95	0.98	1.00	1.00	1.00	1.00	0.96	1.00	0.97	1.00
El. 8	0.99	1.00	1.00	1.00	0.96	1.00	0.99	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00	1.00
El. 9	0.99	1.00	0.99	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00	1.00
El. 10	1.00	0.99	0.96	0.94	0.99	1.00	0.96	1.00	0.99	1.00	0.98	1.00	1.00	1.00	0.98	1.00
El. 11	0.60	0.55	0.70	0.65	0.85	0.60	0.75	0.60	0.55	0.50	0.55	0.50	0.80	0.60	0.65	0.50
El. 12	0.97	0.96	0.89	0.89	0.82	0.93	0.89	0.97	0.96	1.00	0.96	1.00	0.86	0.93	0.93	1.00
El. 13	0.98	1.00	1.00	1.00	0.98	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 14	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 15	1.00	1.00	0.98	0.99	0.97	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.99	1.00	1.00
El. 16	0.98	1.00	1.00	1.00	0.98	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 17	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 18	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 19	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 20	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

It is worth noticing that using a number of particles three time higher does not imply an equivalent growth of the number of iterations (see Table 3), which means that the improvement in the algorithm performance attained by using a bigger population compensates for the increase in computational cost.

Table 3: Number of average FEM simulation for each combination of factors comparing the two levels of population size.

	$C_1 - low$				$C_1 - high$				Average
	$C_2 - low$		$C_2 - high$		$C_2 - low$		$C_2 - high$		
	$w_{min} - low$	$w_{min} - high$	$w_{min} - low$	$w_{min} - high$	$w_{min} - low$	$w_{min} - high$	$w_{min} - low$	$w_{min} - high$	
$N - low$	796	1104	1616	1214	810	840	1460	1024	1108
$N - high$	1428	2628	1884	1800	1326	1584	2070	1074	1724.25
Ratio: high/low	1.79	2.38	1.17	1.48	1.64	1.89	1.42	1.05	1.56

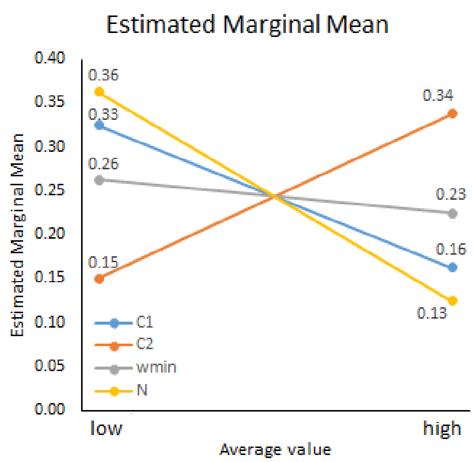

Figure 2: estimated marginal mean

According to the estimated marginal mean (Figure 2) the best combination is the one with high levels for all the factors but the C_2.

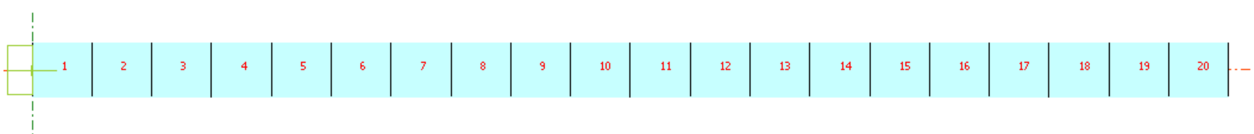

Figure 3: Numbering of the beam's elements.

Finally, this optimized algorithm instance is tested over other damage scenarios applied to the same steel beam. For ease of reference, the numbering of the elements is shown again in Figure 3. In Table 4 all the tests carried out are summarized along with the relevant damage scenarios. The main objective of this final task is to analyse the algorithm performance by simulating the most expected damage conditions occurring in a clamped-clamped beam, namely damages close to the clamps and damage in the mid-point. Furthermore, the tests include specific conditions that, according to experience, complicate the damage identification process, such as: (1) asymmetric configuration of damage, (2) very weak damage extent and (3) multi-damage scenarios.

Table 4: Number of tests and damage scenarios used to analyse the performance of the optimized PSO algorithm.

Test	Damage ratio/Damage location	Description
Par. setting	0.5/Element 11	Relevant damage in the midpoint, asymmetric condition
1st test	0.8/Element 11	Weak damage in the midpoint, asymmetric condition
2nd test	0.95/Element 11	Very weak damage in the midpoint, asymmetric condition
3rd test	0.5/Element 2	Relevant damage close to the clamp, asymmetric condition
4th test	0.5/Element 2 – 0.5/Element 20	Relevant damages close to both the clamps, asymmetric condition
5th test	0.8/Element 2 – 0.5/Element 20	Relevant and weak damage close to both the clamp, asymmetric condition
6th test	0.5/Element 10 – 0.5/Element 11	Relevant damage in the midpoint, symmetric continuous condition
7th test	0.8/Element 1 – 0.8/Element 20	Weak damages close to both the clamps, symmetric condition
8th test	0.8/Element 1 – 0.9/Element 11 – 0.6/Element 20	Mixed damage condition close to both the clamps and in the midpoint

Table 5: Result of the tests: performance indicators.

Test	1	2	3	4	5	6	7	8
Success ratio	1	0.8	1	0.4	0.6	0.4	0.2	0
Fitness	0.0E+00	1.2E-05	0.0E+00	4.1E-04	3.7E-06	1.3E-05	1.2E-03	2.7E-05
N° It	18	34.8	21.6	85.2	65.6	84.2	87.6	101
N° op	1080	2088	1296	5112	3816	5052	5256	6060

Asymmetric damages on a symmetric structure can be easily mistaken when only frequencies are used in the objective function, therefore this kind of problem instances test the sensitivity of the considered features to the damage. Nevertheless, detecting damages with very low extent is essential to test the early warning capability of the algorithm. Usually, identifying weak extent damages as well as multi-damage scenarios are complicated tasks and many algorithms fail in this regard.

Table 5 to 8 report all the results of the numerical experiments. The three single-damage scenarios do confirm the efficiency of the algorithm. Dealing with a very weak damage, the algorithm only fails in one case out of five and the average error is almost negligible. Increasing the number of damages increases the number of false positive errors, despite almost negligible in all the cases. This is due to a common issue of PSO that usually shows a quick convergence in the surrounding of the best solution followed by a slower fluctuation around it. In Figure 4, the convergence trend of one of the unsuccessful runs of damage scenario 6 is showed as an example. In this specific instance, increasing the maximum number of iterations is likely to lead to the correct identification in most cases, but in general the local search capabilities of the algorithm should be improved.

Table 6: Result of the tests: damage extent in each element (tests 1 to 4). In grey correctly identified damages. In yellow false positive errors. Exp and Num are the actual damage and the average results in 10 runs, respectively.

Test	Exp 1	Num 1	Exp 2	Num 2	Exp 3	Num 3	Exp 4	Num 4
El. 1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.96
El. 2	1.00	1.00	1.00	1.00	0.50	0.50	0.50	0.60
El. 3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 4	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 5	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 7	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 9	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 10	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 11	0.80	0.80	0.95	0.96	1.00	1.00	1.00	1.00
El. 12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 13	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 14	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 15	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 16	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 17	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 18	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 19	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.93
El. 20	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.51

Table 7: Result of the tests: damage extent in each element (tests 5 to 8). In grey correctly identified damages. In yellow false positive errors. Exp and Num are the actual damage and the average results in 10 runs, respectively.

Test	Exp 5	Num 5	Exp 6	Num 6	Exp 7	Num 7	Exp 8	Num 8
El. 1	1.00	1.00	1.00	1.00	0.80	0.89	0.80	0.80
El. 2	0.80	0.79	1.00	1.00	1.00	0.84	1.00	1.00
El. 3	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00
El. 4	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00
El. 5	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00
El. 6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 7	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 9	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 10	1.00	1.00	0.50	0.50	1.00	1.00	1.00	1.00
El. 11	1.00	1.00	0.50	0.50	1.00	1.00	0.90	0.91
El. 12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 13	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 14	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 15	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00
El. 16	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00
El. 17	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00
El. 18	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
El. 19	1.00	1.00	1.00	1.00	1.00	0.85	1.00	0.95
El. 20	0.50	0.50	1.00	1.00	0.80	0.89	0.60	0.61

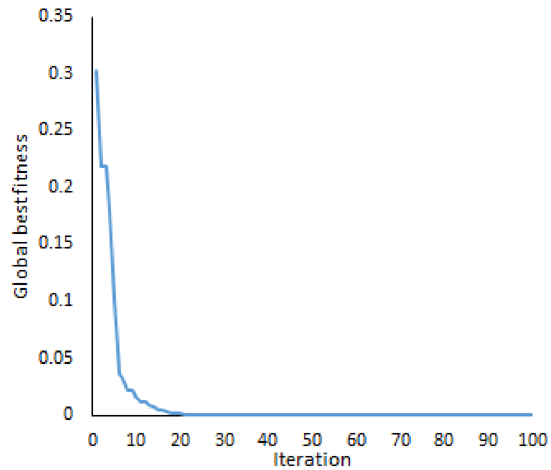

Figure 4: Global best fitness of the swarm along the iterations in test 6 run 3 resulted in a failure (suboptimal solution).

5. Conclusion and future scopes

In the present paper, one of the most basic and well-known version of the PSO algorithm by Shi and Eberhart (Shi & Eberhart, 1998), is used to identify the location and the extent of damage scenarios in a clamped-clamped steel beam. The reference beam is numerically simulated and the damage is introduced in the model through a reduction of the Young modulus. The simulated scenarios are meant to reproduce the most expected damage conditions in the reference beam, namely damage close to the mid-point and damage at the beam clamps.

The analysis allows to confirm that even a basic version of the PSO is suitable for damage identification, although such a version is generally considered not efficient enough in the literature, thereby leading to prefer improved or hybrid versions of the PSO. The influence of parameter setting on the algorithm performance is also confirmed, especially in regard to the coefficients C_1 and C_2, whose values have been usually based on previous works performed on completely different classes of problems. Therefore, it is clear from the developed work how a proper parameter setting is pivotal to achieve an improvement in this field of research.

Beside the aforementioned aspects, the influence of the population size on the algorithm performance is analysed as well. A test involving two levels of the population size demonstrated that using a threefold greater population does not imply an equivalent growth of the number of FEM analyses nor of the time required for the process.

Finally, the optimized algorithm instance resulting from all these analyses is tested over a set of more complex problems. The experiments carried out demonstrated that the PSO is a feasible way to face inverse problems for damage identification, but a few questions are still open and worth of more research.

Real world applications do actually differ from the problem instances used in this study, essentially in two main aspects. First, the features extracted from the monitored structure. Here, such features have a perfect precision, whereas in real world the signal is always polluted by noise. Second, the size of the case study. The beam-like example used in this work is very small compared to real world systems. Therefore, it can be concluded that testing the PSO algorithm over bigger problem spaces, also considering polluted features, is essential. Finally, the robustness of the algorithm should be assessed with yet less information (e.g. number of modes and DOFs) and without disregarding eventual modelling errors (e.g. differences between reference and numerical models in terms of geometry or material properties).

Acknowledgments

This work was supported by FCT (Portuguese Foundation for Science and Technology), within ISISE, scholarship SFRH/BD/115188/2016. This work was also financed by FEDER funds through the Competitivity Factors Operational Programme - COMPETE and by national funds through FCT – Foundation for Science and Technology within the scope of the project POCI-01-0145-FEDER-007633.

References

Adenso-Díaz, B., & Laguna, M. (2006). Fine-Tuning of Algorithms Using Fractional Experimental Designs and Local Search. *Operations Research*, *54*(1), 99–114. https://doi.org/10.1287/opre.1050.0243

Clerc, M., & Kennedy, J. (2002). The particle swarm - Explosion, stability, and convergence in a multidimensional complex space. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, *6*(1), 58–73. https://doi.org/10.1109/4235.985692

Farrar, C. R., & Worden, K. (2007). An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *365*(1851), 303–315. https://doi.org/10.1098/rsta.2006.1928

Gerist, S., & Maheri, M. R. (2016). Multi-stage approach for structural damage detection problem using basis pursuit and particle swarm optimization. *Journal of Sound and Vibration*, *384*, 210–226. https://doi.org/http://dx.doi.org/10.1016/j.jsv.2016.08.024

Kang, F., Li, J., & Xu, Q. (2012). Damage detection based on improved particle swarm optimization using vibration data. *Applied Soft Computing*, *12*(8), 2329–2335. https://doi.org/http://dx.doi.org/10.1016/j.asoc.2012.03.050

Kaveh, A., Javadi, S. M., & Maniat, M. (2014). Damage assessment via modal data with a mixed particle swarm strategy, ray optimizer, and harmony search. *Asian Journal of Civil Engineering*, *15*(1), 95–106. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84884562467&partnerID=40&md5=1eb9e881fde2f86c695211db900d27e5

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *IEEE International Conference on Neural Networks - Conference Proceedings* (Vol. 4, pp. 1942–1948). Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-0029535737&partnerID=40&md5=e6bf04ae50f3268ae545d88ed91d1fc5

Nanda, B., Maity, D., & Maiti, D. K. (2014). Crack Assessment in Frame Structures Using Modal Data and Unified Particle Swarm Optimization Technique. *Advances in Structural Engineering*, *17*(5), 747–766. https://doi.org/10.1260/1369-4332.17.5.747

Seyedpoor, S. M. (2012). A two stage method for structural damage detection using a modal strain energy based index and particle swarm optimization. *International Journal of Non-Linear Mechanics*, *47*(1), 1–8. https://doi.org/https://doi.org/10.1016/j.ijnonlinmec.2011.07.011

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)* (pp. 69–73). https://doi.org/10.1109/ICEC.1998.699146

Vakil Baghmisheh, M. T., Peimani, M., Sadeghi, M. H., Ettefagh, M. M., & Tabrizi, A. F. (2012). A hybrid particle swarm–Nelder–Mead optimization method for crack detection in cantilever beams. *Applied Soft Computing*, *12*(8), 2217–2226. https://doi.org/http://dx.doi.org/10.1016/j.asoc.2012.03.030