Meccanica
DOI 10.1007/s11012-009-9264-5

SIMULATION, OPTIMIZATION & IDENTIFICATION

Application of particle swarm optimization and genetic algorithms to multiobjective damage identification inverse problems with modelling errors

Ricardo Perera · Sheng-En Fang · Antonio Ruiz

Received: 26 February 2009 / Accepted: 26 October 2009

Abstract Structural health monitoring has become an important research topic in conjunction with the damage assessment of structures. The use of system identification approaches for damage detection using inverse methods has become more widespread in recent years and their formulation in a multiobjective framework has become more usual. Inverse problems require the use of an initial baseline model of the undamaged structure. Modelling errors in the baseline model whose effects exceed the modal sensitivity to damage are critical and make an accurate estimation of damage impossible. Artificial intelligence techniques based on genetic algorithms are used increasingly as an alternative to more classical techniques to solve this kind of problem especially due to their feasibility for managing multiobjective problems. This paper outlines an understanding of how particle swarm optimization methods operate in damage identification problems based on multiobjective FE updating procedures and takes modelling errors into account. One experimental example is used to show their performance in comparison with genetic algorithms.

R. Perera (✉) · S.-E. Fang
Department of Structural Mechanics,
Technical University of Madrid, Madrid, Spain
e-mail: perera@etsii.upm.es

A. Ruiz
Department of Applied Mathematics,
Technical University of Madrid, Madrid, Spain

Keywords Multiobjective optimization · Damage identification · Modelling errors · PSO · SPGA

1 Introduction

Over the last few years, there have been increasing demands to develop structural health monitoring systems over different kinds of aerospace, mechanical and civil engineering structures because of the huge economic and life-safety benefits that such technologies have the potential to provide. Vibration testing is the widest used method for dynamic parameter identification in structures (Doebling [1], Yang et al. [2]). The basic idea of these techniques is that modal parameters are functions of the physical properties of the structure. Therefore, using this approach, structural damage can be detected through the correlation between changes in tests data and structural properties since a change in dynamic behaviour will induce a change in modal parameters. Most existing dynamic identification parametric methods make use of frequencies and mode shapes (John et al. [3], Ren and De Roeck [4, 5], Perera and Ruiz [6], Fang et al. [7], Panigrahi et al. [8]) and are based on the methods of model updating. To apply these kinds of methods, one objective function measuring the fit between measured and model predicted data is chosen. Then, optimization techniques are used to find the optimal values of the model parameters that minimize the value of the objective function, i.e., those values best fitting the experimental data

Published online: 11 November 2009

(Brownjohn [9], Friswell [10]). Sometimes, the lack of a clear objective function in the context of real-world damage detection problems recommends simultaneous optimizations of several objectives with the purpose of improving the robustness and performance of the procedure [11–13]. Unlike single-objective optimization problems which accept one single optimum solution, multiobjective optimization problems do not have a single optimal solution, but rather a set of alternative solutions, named the Pareto front set, which contributes to increasing the complexity of the problem and, therefore, special optimization techniques are needed to solve the problem.

Damage identification methods involve fitting a linear physics-based model or baseline model to the measured data from both the healthy and potentially damaged structure. Often these models do not have the fidelity to accurately represent boundary conditions and structural component connectivity, prime locations in which damage can accumulate, especially for large and complex civil structures. This issue together with others can prevent their use in most real-world applications and, therefore, poses a restriction on their applicability. Modelling errors in the baseline model whose effects exceed the modal sensitivity to damage are critical and make an accurate estimation of damage impossible. Damage detection methods for dealing with the effect of the modelling errors have been put forward in recent years [14–18].

Artificial intelligence is a term that in its broadest sense would indicate the ability of a machine to perform the same kind of functions that characterize human thought. Artificial intelligence techniques are becoming useful and popular as alternative approaches to conventional techniques, which is demonstrated by their use for solving complicated practical problems in different sectors, such as engineering, economics, medicine, military, marine, etc. They have also been applied for modelling, identification, optimization, prediction, etc.

Artificial intelligence consists of several branches among which genetic algorithms (GAs) and swarm intelligence are found.

Genetic algorithms, envisaged by Holland [19] in the 1970s, are implemented as a computerized search and optimization procedure that uses the principles of natural genetics and natural selection. GAs consider many points in the search space simultaneously and have been found to provide a rapid convergence to a near optimum solution in many types of problems; they usually exhibit a reduced chance of converging to local minima. Furthermore, as they search for a set of solutions in parallel, unlike conventional optimization techniques, they are particularly appropriate for solving multiobjective problems since the search process can be driven towards a family of solutions representing the set of Pareto optimal solutions. The structure of GAs has been recognized to be more appropriate to multiobjective optimization problems from early on in their development [20] and, furthermore, they lead, in most cases, to the global optimal Pareto front.

Particle swarm optimization (PSO) [21] is one of the newest techniques within the family of optimization algorithms and is based on an analogy with the choreography of flight of a flock of birds. The PSO algorithm relies only on two simple PSO self-updating equations whose purpose is to try to emulate the best global individual found, as well as the best solutions found by each individual particle. Since an individual obtains useful information only from the local and global optimal individuals, it quickly converges to the best solution. PSO has become very popular because of its simplicity and convergence speed.

However, although several extensions to the PSO for handling multiple objectives have been proposed [22, 23], due to their single-point-centred characteristic, it is difficult to locate the non-dominated points on the Pareto front since there will be more than one criteria to direct the velocity and position of an individual. Therefore, certain modifications would be necessary to make PSO suitable for multiobjective problems.

This paper is focused on the study and comparison of both methodologies, GAs and PSO, as well as the proposal of new variations when applied to solve multiobjective identification problems using modal data for assessing structural damage that are based on finite elements updating procedures and take modelling error into account. A successful result would represent an important advance in the real world application of the updating procedures for damage identification since it would combine a procedure that is less sensitive to modelling errors, unavoidable in most real world structures, formulated in a multiobjective framework, more suitable in the context of damage detection problems, and solved using effective and efficient algorithms. Evidently, a key requirement to reach this goal is a suitable formulation of the objective functions dependent on parameters that are sufficiently insensitive to modelling errors but sensitive enough to detect small damage.

This work is structured into five sections. The first section is concerned with the formulation of the multiobjective damage identification problem. The second section is an overview of GA methodology and its extension to multiobjective problems. In the following section some background information about PSO is given. The fourth section describes the proposed improvement combining GAs and PSO concepts in multiobjective damage identification problems in order to enable a faster convergence of the algorithm. Finally, before conclusions, in the last section of the paper the feasibility of the proposed approaches and their comparison is shown in a real damage identification problem.

2 Formulation of the model updating method for damage identification

Damage identification techniques are often based on the methods of model updating which rely on a parametric model of the structure and the minimization of some objective function based on the error between the measured data and the predictions of the model. Using this procedure the optimization process is controlled through a wide range of physical parameters to be updated. Three important aspects for the success of the updating procedure are reliance on the finite element model, including the parameterization of the damage candidate, the choice of the objective function and the solution of the optimization problem.

2.1 Modal parameters non-sensitive to modelling errors

Errors in the baseline finite element model can be very critical for a damage identification procedure. If the effects of the modeling errors exceed the modal sensitivity to damage, accurate damage estimation may not be carried out since the method will have great difficulty in distinguishing between the actual damage sites and the location of errors in the original model. Although updating the modal baseline using the measured data will contribute to improving estimation accuracy, it will not be easy to perform in most real-world structures, which are large and complex, since a lot of measurement information would be needed. For this reason, the objective function to be minimized in the updating procedure should be dependent on modal parameters that are the least sensitive possible to modelling errors in the baseline finite element model and the most sensitive possible to damage.

To solve this problem, the starting point is the equations governing free vibration for undamaged and damaged structures, which in the case of no modelling errors take the following form.

$$([K]-\lambda_j[M])\{\Phi_j\}=0, \quad j=1,\dots,N_m \tag{1}$$

$$[([K]+\Delta[K])-(\lambda_j+\Delta\lambda_j^d)[M]] \times (\{\Phi_j\}+\Delta\{\Phi_j^d\})=0 \tag{2}$$

where $[K]$ and $[M]$ are the global stiffness and mass matrices, respectively, λ_j is the square of the natural frequency corresponding to vibration mode $\{\Phi_j\}$ and N_m is the total number of mode shapes and frequencies considered. Damage is the result of small changes in stiffness $\Delta[K]$ which will lead to a change in frequencies $(\Delta\lambda_j^d)$ and mode shapes $(\Delta\{\Phi_j^d\})$. In the study, it has been assumed proportional damping and that no alteration occurs on the mass matrix before and after damage, which is acceptable in most real applications.

In the case of existing modelling errors of value $\Delta[\tilde{K}]$ in the baseline model, (1) and (2) should be written as follows:

$$[([K]+\Delta[\tilde{K}])-(\lambda_j+\Delta\tilde{\lambda}_j)[M]] \times (\{\Phi_j\}+\Delta\{\tilde{\Phi}_j\})=0 \tag{3}$$

$$[([K]+\Delta[\tilde{K}]+\Delta[K])-(\lambda_j+\Delta\tilde{\lambda}_j+\Delta\tilde{\lambda}_j^d)[M]] \times (\{\Phi_j\}+\Delta\{\tilde{\Phi}_j\}+\Delta\{\tilde{\Phi}_j^d\})=0 \tag{4}$$

where the symbol '$\sim$' denotes the modal quantities due to this error. In (3) and (4) the modelling errors have been included in both damaged and undamaged structures.

After the manipulation of (1), (2), (3) and (4), by neglecting higher order terms of Δ and premultiplying by $\{\Phi_j\}^T$, the following conclusion is obtained:

$$\Delta\lambda_j^d \approx \Delta\tilde{\lambda}_j^d \approx \frac{\{\Phi_j\}^T\Delta[K]\{\Phi_j\}}{\{\Phi_j\}^T[M]\{\Phi_j\}} \tag{5}$$

i.e., the variations of the squares of frequency due to the same change in stiffness $\Delta[K]$ for two FE models with and without a modelling error are approximately equal. That means that the differences of the squares of the frequencies between before and after damage are less sensitive to modelling errors than the frequencies themselves.

On the other hand, from (2), (4) and (5) the following equation can be obtained:

$$([K]-\lambda_j[M])(\Delta\{\Phi_j^d\}-\Delta\{\tilde{\Phi}_j^d\})\approx 0 \qquad j=1,\ldots,N_m \tag{6}$$

whose solution is:

$$(\Delta\{\Phi_j^d\}-\Delta\{\tilde{\Phi}_j^d\})\approx 0, \quad j=1,\ldots,N_m \tag{7}$$

From (5) and (7), it is clear that the differences between the damaged and undamaged squares of the frequencies and mode shapes due to the same change in stiffness $\Delta[K]$ for two FE models with and without a modelling error are approximately equal. This means that these parameters are less sensitive to modelling errors than the frequencies and the mode shapes themselves and, therefore, they might be suitable to formulate a suitable objective function that is scarcely sensitive to modelling errors, in order to solve the inverse problem of damage detection. In this way, any error in the undamaged model of the structures that is also present in the damaged structure will be removed. This does, however, rely on the structure remaining unchanged, except for the damage between the two sets of measurements.

According to the previous paragraphs, for the application of the damage identification method, the baseline model free of modelling errors will be associated with the real structure whose modal parameters have been experimentally measured before and after damage, while the numerical model with modelling errors will correspond to the numerical FE model not updated for intact real conditions before applying the process of damage evaluation.

2.2 Parameterization of damage

One of the key aspects of a model-based identification method is the parameterization of the damage. The effect of damage on the finite element model is introduced according to Continuum Damage Mechanics; in order to do this, damage is quantified through a scalar variable or index d whose values are between 0 and 1. A zero value corresponds to no damage while values next to one imply rupture. In the context of discretized finite elements, the definition of a damage index d_e for each element e allows estimating not only the damage severity but also the damage location since damage identification is then carried out at the element level. These damage indices will be incorporated into the finite element formulation of the studied problem through the stiffness of each element and will represent the relative variation of the damaged element stiffness, k_e^d, to the initial value, k_e:

$$d_e = 1-\frac{k_e^d}{k_e} \tag{8}$$

In this way, the damage indices take part in the characteristic equations of the problem ((1), (2), (3) and (4)) and are the parameters to be estimated during the updating procedure with the purpose of determining the real state of the different parts of the structure.

Although using element stiffnesses as model parameters is a very simple form of equivalent model for the damage, their use is fully justified if the measured modal model consists of the lower natural frequencies and associated mode shapes since, in that case, only a coarse model of the damage can be identified.

2.3 Multiobjective formulation

In the updating procedure, the chosen objective function should reflect the deviation between the numerical model prediction and the real behaviour of the structure. Therefore, the general objective function may be formulated in terms of the discrepancy between the finite element model and experimental quantities. However, according to Sect. 2.1, in order to define an objective function that is scarcely sensitive to modelling errors the function should depend on the changes of the mode shapes and frequencies, both numerical and experimental, instead of mode shapes and frequencies themselves.

From (5) and (7), the optimization problem might be defined by grouping modal frequencies and mode shapes into two groups. The first group would contain all the changes in modal frequencies with the measure of fit selected to represent the mismatch between the changes of the measured and the model-predicted frequencies for all modes, according to (5). The second group, formulated according to (7), would contain the changes of mode shapes with the measure of fit selected to represent the mismatch between the changes of the measured and the model-predicted mode shape components for all modes.

Specifically the two objective functions might be formulated by

$$F_1 = \frac{1}{N_m} \sum_{j=1}^{N_m} \left| \frac{\Delta\lambda_j^d - \Delta\tilde{\lambda}_j^d}{\Delta\lambda_j^d + \Delta\tilde{\lambda}_j^d} \right| \tag{9}$$

$$F_2 = \frac{1}{N_m} \sum_{j=1}^{N_m} \frac{\|\Delta\{\Phi_j^d\} - \Delta\{\tilde{\Phi}_j^d\}\|_2^2}{\|\Delta\{\Phi_j^d\} + \Delta\{\tilde{\Phi}_j^d\}\|_2^2} \tag{10}$$

In a usual damage identification problem with one single objective function, frequency residuals (9) and mode shapes residuals (10) should be combined in one function using suitable weighting coefficients reflecting the relative importance of the objective functions. However, the choice of weights is arbitrary since the importance of each objective is not usually known a priori. However, when combined in a multiobjective framework, no a priori choice has to be performed concerning the relative weighting coefficients scaling mode shape residuals regarding frequency residuals. As opposed to single-objective optimization problems which accept one single optimum solution, multiobjective optimization problems do not have a single optimal solution, but rather a set of alternative solutions, named the Pareto front set, which are optimal in the sense that no other solutions in the search space are superior to them when all objectives are considered. The determination of the set of Pareto-optimal solutions by using traditional optimization techniques requires solving the single objective optimization problem a number of times by varying the weighting factors. However, the methods presented in the following sections have the ability to search for multiple solutions in parallel in a single run and, therefore, the need to use arbitrary weighting factors for weighting the relative importance of each objective is eliminated.

3 Genetic algorithms

Genetic algorithms are an iterative procedure consisting of a constant-sized population of individuals, usually encoded as binary strings and known as chromosomes, representing candidate solutions in a given search space. This space comprises all the possible solutions to the optimization problem at hand. The initial population of individuals is randomly generated. The algorithm evolves over a number a generations through the application of genetic operators until it satisfactorily solves the problem. At every generation, the individuals in the current population are decoded and evaluated according to an objective or fitness function for a given problem. Through the generations, the fitter solutions in the population survive and new individuals are generated which replace the poorer ones.

The genetic operators are used to modify the chosen solutions and select the most appropriate offspring to pass on the succeeding generations with the purpose of finally arriving at a quality solution to the given problem. The three fundamental genetic operators are selection, crossover and mutation. Selection is according to the fitness of individual solutions in such a way that the number of times an individual is selected is approximately proportional to its relative performance in the population. Crossover is performed between two selected individuals by exchanging chromosome segments to form new individuals. Mutation randomly changes part of one selected individual's chromosome and is introduced to prevent premature convergence.

GAs are computationally simple and robust and only examine fitness while ignoring other information such as derivatives. Unlike classical optimization techniques, which adjust a single solution, GAs keep a population of solutions in parallel. This reduces the probability of reaching a false local optimum. Furthermore, this make them ideal for solving multiobjective problems since they are capable of finding several members of the Pareto-optimal set (also called non-dominated solutions) in a single run instead of performing a series of separate runs, which is the case with classical optimization techniques.

3.1 Strength Pareto Genetic Algorithm

A wide variety of multiobjective GAs have been proposed in the literature [23]. Among them, we will give a brief overview of the Strength Pareto Genetic Algorithm (SPGA) which is one of the most representative and will be used in this work.

The SPEA, proposed by Zitzler and Thiele [24] belongs to a second generation of multiobjective evolutionary algorithms born with the introduction of the notion of elitism. In the context of multiobjective optimization, elitism usually refers to the use of an external population to retain the non-dominated individuals found at each generation. For each individual in this external set, fitness is defined through a strength value based on Pareto dominance. This strength is proportional to the number of solutions which a certain individual dominates. For an individual of the non-external set (i.e., the population), its fitness is calculated by adding “1” to the total sum of the strengths of all the external members that dominate it. With this mechanism, population diversity is maintained without any explicit sharing. Additionally, the SPEA also incorporates a clustering procedure in order to keep the size of the external set while maintaining its characteristics. More details about this algorithm can be found in [24].

4 Particle swarm optimization (PSO)

Particle swarm optimization is a population-based stochastic optimization technique developed by Kennedy and Eberhart [21] in 1995 and inspired by the social behaviour of bird flocking and fish schooling. In the same way as genetic algorithms, PSO is initialized with a swarm of random particles and then, using an iterative procedure the optimum is searched. At each iteration, the particles evaluate their fitness (positions relative to the goal) and share memories of their best positions with the swarm. Subsequently, each particle updates its velocity and position according to its best previous position, known as *pbest*, and that of the best particle, known as *gbest*, found so far in the swarm. The updating of the velocity, v_i, of each particle is performed as follows:

$$v_i = w \cdot v_i + c_1 \cdot r_1 \cdot (pbest_i - x_i) + c_2 \cdot r_2 \cdot (gbest - x_i) \tag{11}$$

where w is an inertia coefficient balancing global and local search, r_1 and r_2 are random numbers in the range $[0, 1]$ updated at each generation to help prevent convergence on local optima and c_1 and c_2 are the learning factors which control the influence of $pbest_i$ and *gbest* on the search process. Usually, values equal to 2 are suggested for c_1 and c_2 for the sake of convergence [25]. Additionally, the velocity is limited to a maximum value with the purpose of controlling the global exploration ability of particle swarm and avoiding it moving towards infinity.

This is followed by adjustment of the position, x_i, by:

$$x_i = x_i + v_i \tag{12}$$

The process ends when a specified number of iterations or CPU time is reached or *gbest* cannot be further improved.

For damage identification problems the variable position coincides with the damage indices representing the location and severity of the damage at each element of the FE mesh, which take values between 0 and 1. These constraints can be considered in PSO in a simple, fast and easy way. To do this, during the updating procedure according to (11) and (12), all the particles are kept within the search space, i.e. solutions which do not fall within the valid search space are not accepted. When a decision variable goes beyond its boundaries, it is forced to take the value of the corresponding boundary (either the lower or upper boundary).

The inertia weight w is an important factor for the PSO’s convergence. It controls the impact of the previous history of velocities on the current velocity. A large inertia weight factor facilitates global exploration while a small weight factor facilitates local exploration. Therefore, it is advisable to choose a large weight factor for initial iterations and gradually reduce the weight factor in successive iterations. This can be done by using

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \cdot iter \tag{13}$$

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

As for GAs, PSO algorithms perform nonlinear optimization with a population of potential solutions, so are resistant to trapping in local optima. However, rather than evolving offspring through genetic operators, PSO particles evolve their behaviour (velocities and positions). Because of this, PSO converges to the best solution quickly and is less time consuming than GAs. For success, PSO must balance self-influence and swarm influence. If swarm influence is neglected, poor exchange of information can lead to slow convergence. The influence parameters are often set equal.

5 Multiobjective PSO

Although several extensions to the PSO for handling multiple objectives have been proposed [22, 23], the conventional PSO does not perform well in real-world complex problems like those of damage identification in which the search has to be made in multi-constrained solution spaces. The location of the non-dominated points on the Pareto front will be difficult since more than one criterion will direct the velocity and position of an individual. Because of this, there are many associated problems that require further study for extending PSO in solving multi-objective problems. For example, although the sharing of information among particles based on their previous experiences contributes to increasing the convergence speed, it can be a demerit in multiobjective problems since it reduces the diversity of the algorithm.

The PSO proposed here for multiobjective optimization should aim to preserve population diversity for finding the optimal Pareto front and to include the feature of local fine-tuning for good population distribution by filling any gaps or discontinuities along the Pareto front. Furthermore, the selection process of the best particles with the purpose of extending the existing particle updating strategy in PSO to account for the requirements in multiobjective optimization should be solved too.

Due to the similarity of particle swarm and other evolutionary computation methods, such as genetic algorithms, some multiobjective handling techniques can be adopted to the modified PSO. In this section, the features of the proposed multiobjective particle swarm optimization algorithm (MOPSO) will be described, and details of its implementation will be presented.

5.1 External repository

Following the same guidelines as in SPGA, in the proposal, elitism is implemented in the form of a fixed-size archive or repository to prevent the loss of good particles due to the stochastic nature of the optimization process. The archive is updated at each cycle, e.g., if the candidate solution is not dominated by any members in the archive, it will be added to the archive. Likewise, any archive members dominated by this solution will be removed from the archive.

Furthermore, in order to maintain a set of uniformly distributed non-dominated particles in the archive, the average linkage method [26] has been chosen. The basic idea of this method is the division of the non-dominated particles into groups of relatively homogeneous elements according to their distance. The distance d between two groups, g_1 and g_2, is given as the average distance between pairs of individuals across the two groups

$$d = \frac{1}{|g_1| \cdot |g_2|} \cdot \sum_{i_1 \in g_1, i_2 \in g_2} \|i_1 - i_2\| \qquad (14)$$

where the metric $\|\cdot\|$ reflects the distance between two individuals i_1 and i_2.

Then, following an iterative process, the two groups or clusters with minimal average distance are amalgamated into a larger group until the number of clusters is equal to the maximum permitted capacity of the repository. Finally, the reduced non-dominated set is computed by selecting a representative individual per cluster, usually the centroid. With this approach a uniform distribution of the grid defined by the non-dominated solutions can be reached.

5.2 Selection of *pbest* and *gbest*

In MOPSO, *gbest* and *pbest* play an important role in guiding the entire swarm towards the global Pareto front. Contrary to single objective optimization, the *gbest* and *pbest* for multiobjective optimization exist in the form of a set of non-dominated solutions. The selection of *pbest* follows the same rule as in traditional PSO but applying the Pareto dominance concept.

In MOPSO there is no such thing as the best position vector such as in standard PSO. There are several equally good non-dominated solutions stored in the external repository. The selection of an appropriate *gbest* is critical for the search of a diverse and uniformly distributed solution set. In order to promote diversity and to encourage exploration of the least populated region in the search space, the choice of *gbest* is performed by the roulette wheel selection scheme based on fitness assignment. For its application, all the particles of the external repository are assigned weights based on their fitness values; then the choice is performed using roulette wheel selection. The fitness assignment mechanism for the external population proposed by Zitzler and Thiele [24] has been used in the proposed algorithm. According to this mechanism each individual i in the external repository is assigned a strength s_i proportional to the number of swarm members which are dominated by it, i.e.

$$s_i = \frac{n_i}{N + 1} \qquad (15)$$

where n_i is the number of swarm members which are dominated by individual i and N is the size of the swarm. The fitness value of the individual i is the inverse of its strength. Therefore, non-dominated individuals with a high strength value and, therefore, located in regions more densely populated are penalized and have a lower probability of being selected.

5.3 Mutation operator

PSO is known to have a very high convergence speed. However, in the context of complex multiobjective optimization problems, which are usual for many real world problems, such a convergence speed may result in a premature convergence to a false Pareto front due to the difficulty of incorporating the heuristic characteristic of complex problems into the algorithm by using only the two PSO equations. This phenomenon might be avoided by encouraging the exploration to regions beyond those defined by the search trajectory. This is the main motivation for using a mutation operator. High convergence speed offered by the PSO justifies the use of a mutation operator, which delays convergence and enables the particles to explore the whole search space.

By defining a low constant mutation probability the exploration ability in searching solution space is increased, without excessively increasing the diversity of the population, thereby avoiding premature convergence problems.

6 Experimental validation of SPGA and MOPSO

To check the true robustness and efficiency of the proposed methods a real one storey and one bay RC frame experimentally tested in the Structures Laboratory of the Structural Mechanics Department of Madrid Technical University (Spain) has been considered. The geometric dimensions and the reinforcement layout in the sections are illustrated in Fig. 1 and the material properties were assigned as follows: (a) for concrete, the elastic modulus, compressive strength and mass density were taken to be $E = 34093$ MPa, $f_c = 32$ MPa and $\rho = 2300$ kg/m^3; (b) for steel reinforcement, the elastic modulus, elastic limit and mass density were taken to be $E = 210000$ MPa, $f_y = 510$ MPa and $\rho = 7850$ kg/m^3, respectively.

In the test programme, the specimens are subjected to an alternative combination of increasing static load

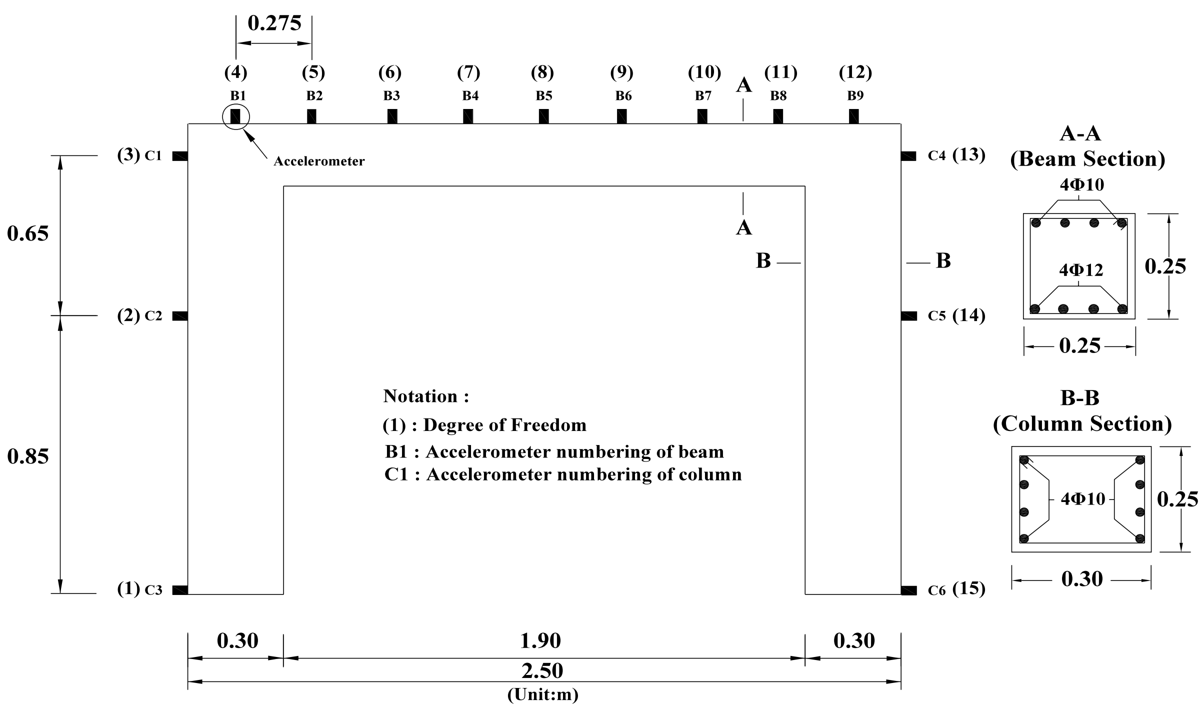

Fig. 1 Geometry and reinforcement of the RC frame and dynamic test set-up

tests and dynamic tests. The first are performed to gradually introduce the cracks in the specimens while the second allow determining the dynamic characteristics after each static test. In this way, it is possible to obtain the dynamic parameters of the specimen for different loading stages and, therefore, for different levels of cracking.

Static tests were performed using a closed-loop hydraulic actuator mounted vertically on a steel frame. In modal testing, the frame was excited by an impact hammer and the response was monitored with 15 accelerometers whose locations are shown in Fig. 1. For the modal testing, the RC frame was supported on two rubber springs at the column bases in order to simulate a free boundary condition.

An initial load was applied on the intact structure and then four increasing loading stages were carried out to introduce cracks. During the first two levels of loading, cracks only appeared at the midspan of the beam. However, during the last two levels of loading, cracks also developed at the beam part of the joint.

Frequencies and mode shapes evaluated from modal tests performed over the undamaged initial stage of the frame and over each one of the four damage stages were used in the two objective functions defined in (9) and (10). For the numerical model, the concrete frame was modelled with planar beam elements with two degrees of freedom per node. The number of elements was defined according to the location of the accelerometers (Fig. 1) finally resulting in 12 elements, 2 for each column and 8 for the beam. Furthermore, none of the nodes in the FE model was constrained in order to simulate the free boundary condition close to the experimental boundary condition. However, modelling errors are present in the numerical model since this has not been updated with the experimental results corresponding to the undamaged initial stage of the frame with the purpose of checking the robustness of the proposed approaches.

The two proposed algorithms, SPGA and MOPSO, were used for solving the multiobjective problem of damage detection after each one of the four loading stages. The problem has 12 decision variables coincident with the level of damage at each element of the numerical model. Standard values were selected for the PSO parameters of (11) and (13): Cognitive parameter $c_1 = 2$, social parameter $c_2 = 2$, initial inertia weight $w_{\max} = 0.95$, final inertia weight $w_{\text{end}} = 0.4$, maximum velocity $v_{\max} = 100$, mutation probability $= 0.01$. The chosen parameters for the SPGA are also standard: Crossover probability $= 0.7$; mutation probability $= 0.01$; tournament selection scheme; each design variable or damage variable has been coded into a 3-bit binary number. In both approaches the size of the external archive was set to 30 individuals.

The comparisons were made from results derived after an identical number of solution evaluations were performed, using that factor as a measure of common computational effort. Taking into account the stochastic nature of the algorithms, thirty independent runs were performed to decrease the influence of random effects with the same parameter settings, as previously mentioned. Then as stop criterion the total number of objective function evaluations was set at $10{,}000 \times 30$, i.e., 100 iterations per run, since a size of population of 100 individuals was chosen for both approaches.

When dealing with multiobjective optimization problems, the qualitative assessment of results becomes difficult because several solutions will be generated instead of only one. Different performance measures can be used to evaluate the Pareto fronts produced by the various GAs. These include the size of the Pareto front and its diversity and the coverage of the different Pareto fronts. This last measure allows setting if the outcomes of one algorithm dominate or not the outcomes of another algorithm.

With the conditions mentioned above, the comparison of the Pareto front curves for the MOPSO and SPGA algorithms and for the four loading stages are shown in Figs. 2a–d, with the objective functions F_1 and F_2 on the horizontal and vertical axis, respectively. The graphical representations indicate, at first glance, that, in general, MOPSO performs better than SPGA.

Pareto front size and average density values by the two algorithms are shown in Table 1.

Concerning the coverage of the different Pareto fronts, MOPSO is clearly superior to SPGA since none of the Pareto points of MOPSO is weakly dominated by Pareto points of SPGA for any of the four loading stages, i.e., $C(\text{SPGA}, \text{MOPSO}) = 0$, where $C \in [0, 1]$ measures the coverage such as defined in [27]. On the contrary, Pareto points of SPGA are weakly dominated by Pareto points of SP, specifically $C(\text{MOPSO}, \text{SPGA})$ takes values of 0.64, 0.76, 0.75 and 0.84 for the four loading steps, respectively, where the value $C(\text{MOPSO}, \text{SPGA}) = 1$ means that all the Pareto points derived with SPGA are dominated by or equal to points obtained with MOPSO.

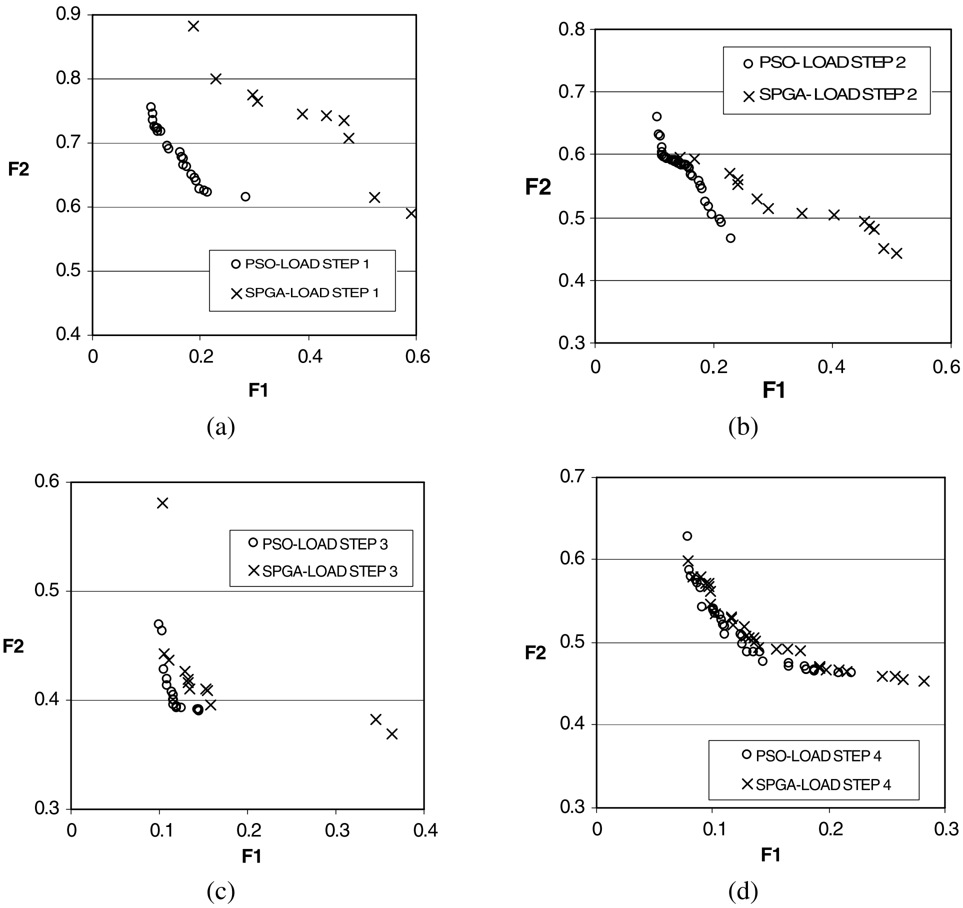

Fig. 2 Pareto fronts: (**a**) Load step 1 (**b**) Load step 2 (**c**) Load step 3 (**d**) Load step 4

Table 1 Average density values and Pareto front sizes

	LOAD STEP 1		LOAS STEP 2		LOAD STEP 3		LOAD STEP 4	
	MOPSO	SPGA	MOPSO	SPGA	MOPSO	SPGA	MOPSO	SPGA
Average Density Value	16.33	3.57	31.42	4.76	16	7.17	22.81	20.48
Pareto front size	22	14	38	17	16	12	31	31

All the results above demonstrate that the performance registered with MOPSO is much better as compared with SPGA. The better quality of Pareto fronts obtained by the proposed MOPSO algorithm as compared with the SPGA algorithm is attributed to the specific combination of established as well as the novel multiobjective tools utilized.

The average processing time of both methods for the same number of evaluations was very similar although with MOPSO the same quality of the Pareto front obtained with SPGA in 100 iterations per run was reached in a considerably minor number of iterations confirming the high convergence speed of the PSO method even when it is modified for its application to multiobjective problems.

Finally, damage predictions for the RC frame with both methods are shown in Fig. 3a and b for the beam part of the frame, which represents its damaged part. For simplicity, only the first and third loading stages are represented since similar conclusions were ob-

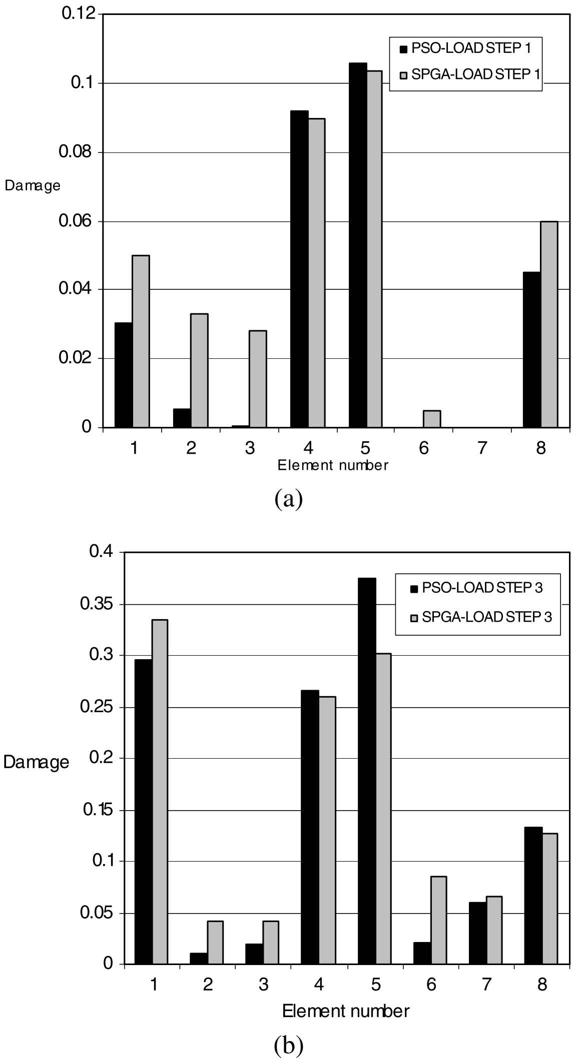

Fig. 3 Damage distribution: (**a**) Load step 1 (**b**) Load step 3

tained for the second and fourth loading stage, respectively. From the set of Pareto optimum solutions obtained for each algorithm, the represented solution is the average of the ones minimizing the following expression

$$\sqrt{F_1^2 + F_2^2} \tag{16}$$

for each run.

In spite of some false warnings, damage distribution agrees with the experimental observations, especially when MOPSO is applied due to the better quality of the solutions obtained. As many cracks at the midspan section of the beam in the first loading stage as cracks close to the joint in the third loading stage are perfectly identified.

7 Conclusions

This paper proposes a PSO method for solving multiobjective damage identification problems based on model updating methods and modal tests when modelling errors are present. The success of the proposed procedure has required firstly the formulation of suitable objective functions dependent on modal parameters scarcely sensitive to modelling errors. Secondly, different modifications have been introduced over the standard PSO algorithm with the purpose of improving its performance when applied to a multiobjective framework. Furthermore, the proposed MOPSO method has demonstrated its superiority when compared to other more established methods like SPGA.

With properties such as fast convergence, storage and continuous modifications of potential solution in external repository, self adaptive mutation, and a simple yet efficient constraint handling methodology, MOPSO can evolve as an alternative and preferable tool for structural damage identification problems. Future research should be directed to exploring the applications and performance of MOPSO in large and complex civil structures in the real world in which the difficulty of formulating reliable baseline models is very high and where the efficiency and effectiveness of MOPSO would not be ensured because of possible premature convergence in local optima due to the high convergence speeds.

Acknowledgement This work is supported by the Ministry of Education and Science of Spain through project BIA2007-67790.

References

1. Doebling SW, Farrar CR, Prime MB (1998) A summary review of vibration-based damage identification methods. Shock Vib Dig 30:91–105
2. Yan YJ, Cheng L, Wu ZY, Yam LH (2007) Development in vibration-based structural damage detection technique. Mech Syst Signal Process 21:2198–2211
3. John EM, Michael IF, Cristinel M (1999) Method for determining model structure errors and for locating damage in vibrating systems. Meccanica 34(3):153–166
4. Ren WX, De Roeck G (2002) Structural damage identification using modal data. I: simulation verification. J Struct Eng ASCE 128:87–95

5. Ren WX, De Roeck G (2002) Structural damage identification using modal data. II: test verification. J Struct Eng ASCE 128:96–104
6. Perera R, Ruiz A (2008) A multistage FE updating procedure for damage identification in large-scale structures based on multiobjective evolutionary optimization. Mech Syst Signal Process 22:970–991
7. Fang SE, Perera R, De Roeck G (2008) Damage identification of a reinforced concrete frame by finite element model updating using damage parameterization. J Sound Vib 313:544–559
8. Panigrahi SK, Chakraverty S, Mishra BK (2009) Vibration based damage detection in a uniform strength beam using genetic algorithm. Meccanica. doi:10.1007/s11012-009-9207-1
9. Brownjohn JMW, Xia PQ, Hao H, Xia Y (2001) Civil structure condition assessment by FE model updating methodology and case studies. Finite Elem Anal Des 37:761–775
10. Friswell M (2008) Damage identification using inverse methods. In: Morasi A, Vestroni F (eds) Dynamic methods for damage detection in structures. Springer, New York, pp 13–66
11. Haralampidis Y, Papadimitriou C, Pavlidou M (2005) Multi-objective framework for structural model identification. Earthquake Eng Struct Dyn 34:665–685
12. Jaishi B, Ren WX (2007) Finite element model updating based on eigenvalue and strain energy residuals using multiobjective optimisation technique. Mech Syst Signal Process 21:2295–2317
13. Perera R, Ruiz A, Manzano C (2007) An evolutionary multiobjective framework for structural damage localization and quantification. Eng Struct 29:2540–2550
14. Yq Ni, Wang BS, Ko JM (2002) Constructing input vectors to neural networks for structural damage identification. Smart Mater Struct 11:825–833
15. Parloo E, Guillaume P, Van Overmeire M (2003) Damage assessment using mode shape sensitivities. Mech Syst Signal Process 17:499–518
16. Titurus B, Friswell MI, Starek L (2003) Damage detection using generic elements: Part I. Model updating. Comput Struct 81:2273–2286
17. Titurus B, Friswell MI, Starek L (2003) Damage detection using generic elements: Part II. Damage detection. Comput Struct 81:2287–2299
18. Lee JJ, Lee JW, Yi JH, Yung CB, Jung HY (2005) Neural-networks-based damage detection for bridges considering errors in baseline finite element models. J Sound Vib 280:555–578
19. Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Harbor
20. Coello CA (2005) Recent trends in evolutionary multiobjective optimization. In: Goldberg AJ (ed) Evolutionary multiobjective optimization: theoretical advances and applications. Springer, London
21. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, Piscataway, NJ, pp 1942–1948
22. Hu X, Eberhart R, Shi Y (2004) Recent advances in particle swarm. In: IEEE congress on evolutionary computation, Portland, Oregon, USA
23. Coello CA, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. IEEE Trans Evol Comput 8:256–279
24. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 3:257–271
25. Eberthart RC, Shi Y (2001) Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 congress on evolutionary computation, Seoul 1, pp 81–86
26. Morse JN (1980) Reducing the size of the nondominated set: Pruning by clustering. Comput Oper Res 7
27. Perera R, Ruiz A, Manzano C (2009) Performance assessment of multicriteria damage identification genetic algorithms. Comput Struct 87:120–127