Crack size identification by means of model reduction using particle swarm optimization and genetic algorithm

Brahim Benaissa, Idir Belaidi
Dept. of mechanical engineering
UMBB
Boumerdes, Algeria

Nourredine Aït Hocine
LMR
INSA Centre Val de Loire
Blois, France

Abstract—**In this study we used the material elastic properties as a base. A tow dimensional cracked plate under traction is modelled by finite element method than a reduced model is built using the proper orthogonal decomposition method, for a cracked plate. The crack length is estimated as an inverse identification problem, basing on the deformation obtained from the boundary nodes of the structure considered as sensor points. A genetic algorithm and particle swarm optimization are used for the minimization of the error function which is expressed as the difference between displacement field of the boundaries caused by the calculated crack size and the field measured at the actual identity. The approach presented accurate results and could guess the real crack size in a precession less that 1E-6 of the cost function. Aiming to help selecting the best optimization method for inverse crack identification problems, the accuracy of both optimization algorithms is put into question as result their performance is evaluated and compared.**

Keywords Particle swarm optimization, genetic algorithm, inverse problem, model reduction, crack size identification.

I. INTRODUCTION

Inverse problems particularly those used for crack detection and crack identification can be stated as an optimization task. They are defined as the problems where the output is known and the input or source of output remains to be determined [1]. In the case of the Inverse Elasto-Statics Problem (IESP) of internal defect identification, and the location, the orientation and the size of the crack are unknown but the displacements along the boundaries are known. In order to identify the crack, its corresponding boundary displacements data are compared with other data related to known cracks.

In inverse identification problems, the optimization algorithm imposes respective computations performed by numerical model that should represent a compromise between high accuracy and short computational time. The boundary element method (BEM) is a well know structural analysis method for its low computational-cost [2,3], while the finite element method (FEM) is relatively a high computational-cost method. Model order reduction insures the advantages of FEM with lower cost [4].

Proper orthogonal decomposition (POD) is an model reduction techniques based on the results of simulations [5]. It has been used for damage identification [6-8]. Radial basis functions (RBF) is a powerful interpolation tools [9]. POD-RBD provide a high accuracy and widely improve the results of the reduced model [10-12].

The term optimization refers in most cases to minimize objective function by choosing values for its variables within a satisfactory range [13]. Particle swarm optimization (PSO) and Genetic algorithm (GA) are popular evolutionary optimization techniques used in mainly every optimization problem. They are suitable for crack identification, contrary to classical methods where the continuity of the objective function is essential. They also present a high possibility of converging to local optimum which is strongly dependent on choice of starting point [14,15].

The main purpose of this study is investigate and compare the performance and the efficiency of the two algorithms GA and PSO when applied to a crack size estimation problem. POD is used to build a reduced model of central cracked plate based on finite element results.

II. MODEL REDUCTION

A. Problem discription

In the case of the elasticity, the crack is represented by a line segment. Thus, the parameters to be studied are the crack's position, its length and orientation angle in the plane. In this work, the position and the orientation angle are fixed in the center of the plate by 0° angle, and the crack's size is targeted.

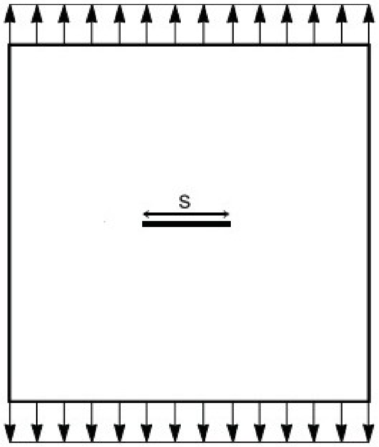

Figure 1: Center crack tension specimen.

A plane strain plate is considered to be the 40 x 40 mm square. The Plate containing a single crack is subjected to a traction load (1mm displacement is imposed on both up and down sides) has been simulated using FE commercial code ABAQUS, where the displacement values of all nodes of external boundaries are collected for the construction of the reduced model, from applications of different crack sizes s belongs to the range 0 (no crack) to 12 mm. According to double symmetries, only a quarter of the plate has been modelled.

B. POD-RBF

POD is a powerful statistical method for data analysis employed as a model order reduction technique in many fields [16]. In our study, the POD is used to build a reduced model of a two dimensional cracked plate under traction, to determine the boundary displacement field corresponding to different crack sizes, based on the results of the finite element simulations, this known as the method of snapshots. The snapshot consists of the displacement vectors of the boundary nodes which are expected to be correlated. They are stored in matrix U.

$$U = \begin{bmatrix} u_1^1 & u_1^2 & \cdots & u_1^S \\ u_2^1 & u_2^2 & & u_2^S \\ \vdots & \vdots & \ddots & \vdots \\ u_N^1 & u_N^2 & \cdots & u_N^S \end{bmatrix}, \qquad (1)$$

where N is the total number of nodes and S is the number of snapshot vectors U_i or FEM simulations results, each one corresponding to a crack length value. Parameters matrix P stores the crack length valuesP_i . The main purpose of POD is to construct a set Φ of orthogonal vectors called POD basis vectors, resembling the snapshot matrix U in an optimal way by exploiting the expected correlation between the results vectors expressed in the linear relationship:

$$U = \Phi \cdot A \qquad (2)$$

Where A is the matrix collecting the coefficients of the new basis combination. It is called the amplitude matrix. Referring to the orthogonality of Φ, it can be computed from:

$$A = \Phi^T \cdot U \qquad (3)$$

Optimal basis vectors are defined by the performance of the proper orthogonal decomposition (POD):

$$\Phi = U \cdot V \cdot \Lambda^{-1/2} \qquad (4)$$

Matrix V stores the normalized eigenvectors of the covariance matrix C and Λ is a diagonal matrix storing its eigenvalues:

$$C = U^T \cdot U \qquad (5)$$

A high accuracy $\widehat{\Phi}$ low dimensional approximation is extracted from Φ constructed as a POD basis. This is accomplished by preserving only K ($K \ll S$) columns of Φ that correspond to the largest eigenvalues; consequently the amplitude matrix $\widehat{A}$ is specified by:

$$\widehat{A} = \widehat{\Phi}^T \cdot U \qquad (6)$$

Since,

$$U = \widehat{\Phi} \cdot \widehat{A} \qquad (7)$$

The use of RBF interpolation different sets of parameters can be generalized not already included in the initial selection P. The amplitudes matrix A is defined by the combination of interpolation functions of the parameter vector P gathered in the matrix G. The matrix B gathers the unknown coefficients of this combination:

$$A = B \cdot G \qquad (8)$$

The interpolation functions are stated by [17]:

$$g_i = g_i(|P - P_i|) = \frac{1}{\sqrt{|P - P_i|^2 + c^2}} \qquad (9)$$

P_i is the parameter corresponding to U_i (for i=1,2,…,S). The argument of the i-th RBF is the distance $|P - P_i|$ between its current parameter P_i and the reference parameter P. c is the RBF smoothing factor. As the vector P is normalized, c is defined in the range from 0 to 1.

After the coefficient matrix B is evaluated, a reduced model of (8) can be put under vector form:

$$a(P) = B \cdot g(P) \qquad (10)$$

By defining the amplitude vector as a function of parameters, the equation (7) can be expressed as the approximation of the snapshot u corresponding to a new parameter vector P:

$$u(P) = \widehat{\Phi} \cdot a(P) \qquad (11)$$

This reduced model is referred to as the trained POD-RBF network. It is completely able to reproduce unknown boundary displacement field of the structure corresponding to any set of crack parameter (length) P. It is noted that extrapolation outside the range of P leads to poor precision of the model. Also if the knot points P_i are very close to each other, the matrix G could be singular, which can be avoided by reducing the c value.

III. Optimization algorithms

A. Genetic Algorithm

The genetic algorithm is an evolutionary optimization method, widely used in last decade [18].

In a genetic algorithm, individuals are the feasible solutions randomly generated in the design space. They evolve toward better solution iteratively in a process inspired from the natural evolution. Each individual has a set of properties represented in binary encoding or other encoding called chromosomes. They are allowed to reproduce and cross among themselves in order to obtain favorable solutions. The fitness is the objective function value, as the objective function plays the role of the environment. The best feasible solutions are given a higher probability of being chosen as parent to new individuals, where the properties of the parents are combined by exchanging chromosomes parts, producing two new designs. The mutation is then preformed on the resulting individuals, by randomly replacing the digits inside a randomly selected chromosome. These basic operators are repeated to create the next generations, until the stopping criterion is satisfied, i.e. generally when reaching a maximum number of generations or when a satisfactory fitness value has been achieved [19].

The identification process is implemented in MATLAB. The chosen genetic parameters are: crossover rate = 0.8 and mutation rate = 0.01.

The specific implementation is structured as follow:

1. Creation of a starting population of N individuals randomly. Each individual has one chromosome, corresponding to the crack length.
2. Evaluation of each individual, that correspond through the reduced model to a boundary displacement vector $\mathrm{u(P)}$, then calculate the fitness value which is the error between the resulting vector and the reference displacement cause by the real crack parameters $\mathrm{u(P_0)}$ expressed as:

$$\begin{cases} \mathrm{F(P)} = \frac{\|\mathrm{u(P_0)} - \mathrm{u(P)}\|^2}{\|\mathrm{u(P_0)}\|^2} \\ \mathrm{F(P_{optimal}\ )} = \min [\mathrm{F(P)}] \end{cases} \quad (12)$$

3. Terminate the algorithm if the stopping criteria is reached. Else continue.
4. Ranking the population according their fitness value. Then select a proportion for reproducing a new generation. The top ranked are favorable to be selected.
5. Performance of the crossover operation.
6. Mutation of an indicated percentage of the resulting individuals.
7. Replacement of the old population by new one and go to step 2.

B. Particle Swarm Optimization

The Particle Swarm Optimization (PSO) is a population- based optimization method inspired from the behavior of bird flocks, which is characterized by distinct social and psychological principles. This method has drawn the wider attention of several researchers, in recent years. It can be used in both simple and large-scale structural optimization problems.

The algorithm was first proposed by Kennedy and Eberhart [20], and has been modified to handle several optimization problems with presence or absence of constraints. PSO requires a relatively small number of parameters, which facilitates the implementation of the algorithm and reduces the computational cost.

The main idea of PSO is that the candidate solution is considered as a particle moving through the design space, searching for the global optimum position. Initiated as a group of random particles, each particle is characterized by its position in the multidimensional space and its speed of movement. The particles cooperate with each other to achieve the solution based on their personal previous experience and that of their colleagues over iterations, by remembering the best position corresponding to the best fitness they had crossed so far called personal best (Pbest) and the best position that any of other particle of the swarm has crossed so far called global best (Gbest). Iteratively the speed of each particle is updated stochastically, then the positions are updated using the new speed vectors.

The speed and the position of the particles are updated as follows:

$$\{\mathrm{v^i(t+1)}\} = \mathrm{w}\{\mathrm{v^i(t)}\} + \mathrm{c_1}\{\mathrm{r_1}\} \cdot \left(\{\mathrm{x^{Pb,j}}\} - \{\mathrm{x^j(t)}\}\right) + \mathrm{c_2}\{\mathrm{r_2}\} \cdot \left(\{\mathrm{x^{Gb}}\} - \{\mathrm{x^j(t)}\}\right) \quad (13)$$

$$\{\mathrm{x^i(t+1)}\} = \{\mathrm{x^i(t)}\} + \mathrm{v^i(t+1)} \quad (14)$$

The weight inertia parameter w is multiplied by the value of the particle speed at every iteration to control the acceleration of the particle in its original direction.

Equal to 0.95 in this application, it can also be updated during iterations. $\mathrm{c_1}$ is a positive constant called cognitive parameter and controlling the step size toward the particle's personal best position. c_2 is the social parameter that control the step size toward the global best position. In this study $c_2 = c_1 = 2$. $\{r_1\}$ and $\{r_2\}$ are vectors containing random numbers within the interval [0,1]. $\{x^j(t)\}$ is the vector of the current positions of particles. $\{x^{Pb,j}\}$ is the vector of the personal best position found by the particle j. $\{x^{Gb}\}$ is the vector of the global best position found by the entire swarm.

The algorithm's structure is described as follows:

1. Initialization of the algorithm by randomly generating the particle's position vectors within the design space and calculating their corresponding speed vectors.
2. Evaluation of the fitness value for current positions of the particles corresponding to boundary displacement vector $u(P)$ given by POD-RBF (Eq. (12)).
3. Personal best setting: for each j particle, set the current position as the new xPb,j. If the current fitness value is better than the best fitness value in the particle's history, it is set as the new Pbest.
4. Global best setting: set as Gbest the best fitness value of all the particles.
5. Updating: Calculate particle velocity from Eq.(13) and update particle position from Eq. (14).
6. Feasibility check: for any dimension i, if $x_i \leq x^{min}$ or $x_i \geq x^{max}$ then $x_i = x^{min}$ and $x_i = x^{max}$ respectively and $v_i = 0$.
7. Ending: If the maximum number of iterations or a defined fitness value is reached, the algorithm is terminated; else, the steps 2 to 6 are repeated.

IV. Results and Discussion

The inverse problem is formulated as a comparison between two vectors; one represents the boundary displacement proposed by the optimization processes at each evaluation, the second is the reference displacement field caused by the crack we want to predict its size. It is solved by GA and PSO.

Because both methods are based on random initiation, 5 applications of crack size estimation are made for real crack size equal 2 mm, giving different results in each application. A number of 100 iteration (generations for GA) and fitness value of 1E-06 are considered as a stopping criteria. Population size and particles number is equal to 10.

A. Preformence of the identification algorithms

Figure 2 and Figure 3 depicts respectively the fitness convergence and crack size convergence of GA and PSO from five applications. Table 1 illustrates the results issued from GA and PSO, where minimum iterations is the minimum number of iteration taken until reaching the convergence criteria. Maximum iterations is the highest number of iteration taken until the stopping criterion is reached. The standard deviation of the crack size result is calculated to show the sharpness of the methods. The average number of iterations and average time are measured from all five applications showing the power of each method.

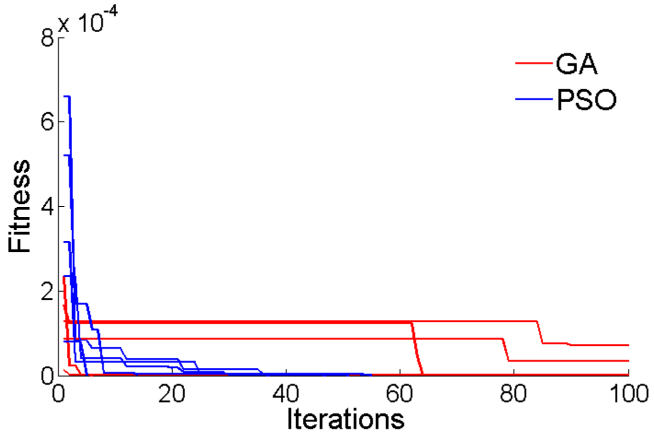

Figure 2: Fitness convergence of GA and POD

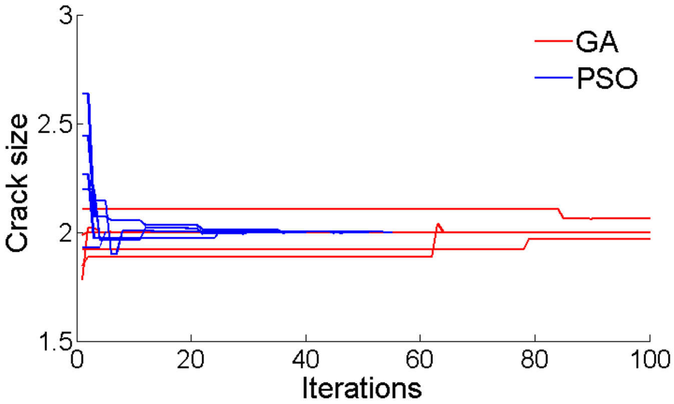

Figure 3: Estimated size convergence for GA and POD

Table 1: Crack size identification results from GA and PSO applications.

	Maximum iterations	Minimum iterations	average number of iterations	average time (sec)	Best fitness	Best result (mm)	standard deviation
GA	100	6	73.8	430.8	2.44E-07	2.000207	0.0335
PSO	55	5	39	228.8	3.66E-08	2.000102	0.0003

The results demonstrate that both algorithms provides results close to real crack size, but it is clearly shown the PSO technique is more adoptable for the crack size estimation problem, as it is very quick compared to GA, it take less iterations. PSO have successfully led to the real solution at every application, before the 55th iteration, unlike the GA which seemed to fail to attend the solution before the 100th iteration.

It is noted that the PSO insure the convergence to the real result in every application which is expressed by their standard deviation equal to 0.031% which is far less than the one calculated from GA result 3.35%.

In terms of computational time, there is big advantage for PSO when comparing the average needed time for all application. The GA tends to stagnate depending on the closeness of fist generation to the real solution. There is no big difference in minimum iteration number; 5 iterations by PSO and 6 generations by GA, it took 31.9 second and 43.8 second respectively.

B. Noise effect

In order to study the stability of the inverse identification to measurements noise, four levels of perturbation has been added to the exact input deformation vector, for crack with size equal to 10 mm. Table. 2 illustrate the performance of both optimization algorithms for the noise levels: 1%, 2% 5% and 10% respectively, compared to the results of a noise-free application (0%). The noise is determined by the White Gaussian law. Noise level is indicated by a percent value which is linked to the standard deviation of the noise. In the table, equivalent to each noise level, the results of crack length and its corresponding fitness value are presented respectively.

TABLE 2: testing noise effect on the accuracy of the optimization methods for crack size equal to 10 mm.

Noise	GA	PSO
0%	9.993 ; 2.2E-5	9.999 ; 7E-8
1%	10.005 ; 1.5E-4	9.986 ; 5.5E-5
2%	10.059 ; 2.6E-4	9.974 ; 1.1E-4
5%	10.894 ; 3.4E-3	9.931 ; 2.6E-4
10%	9.862 ; 5.4E-4	9.864 ; 5.4E-4

The variations obtained in the crack identity are in good agreement with the noise levels. It is noted that 1% of noise level does not affect the results, both optimization methods presented good accuracy, with a slightly difference for the advantage of PSO approach, it is in ahead of GA by giving a minimum fitness value at each level.

V. CONCLUSION

In this paper a comparative study has been made using Particle Swarm Optimization and Genetic Algorithm for crack size identification by mean for model reduction. Overall the simulation results indicate that both GA and PSO can be used to identify the size parameters accurately. With respect to minimizing the objective function Integral Square Error, the PSO determines a minimal value than does the GA. In terms of computational time, the PSO approach is faster than GA, although it is noted that neither algorithm takes what can be considered an acceptable time to determine the results. The stability of both optimization algorithms to measurement noise is tested by introducing a white Gaussian noise.

The algorithms like GA and PSO are inspired by nature, and it has been proved that they lead to effective solutions in optimization problems. These techniques possess apparent robustness. There are various control parameters, and appropriate setting of these parameters is a key point for success. The possibility of performing hybrid approaches should be considered. Additionally for both approaches the major issue in implementation is based on the selection of an appropriate objective function.

REFERENCES

[1] S. Kubo, Inverse analyses and their applications to nondestructive evaluations, in Proc. 12th A-PCNDT - Asia-Pacific Conference on NDT, Auckland, ,2006.

[2] S. C. Mellings and M. H. Aliabadi, "Flaw identification using the Boundary Element Method", International Journal For Numerical Methods In Engineering, vol. 38, 399-419,1995.

[3] N. Amoura, H. Kebir, S. Rechak, J. Roelandt, Axisymmetric and two-dimensional crack identification using boundary elements and coupled quasi-random downhill simplex algorithms, Engineering Analysis with Boundary Elements, 611-618, 2010.

[4] W.H. Schilders, H.A. Van Der Vorst, J. Rommes, Model order reduction: theory, research aspects and applications, Springer, 2008.

[5] A. Chatterjee, An introduction to the proper orthogonal decomposition, Current science, 808-817, 2000.

[6] F. Lanata, A. Del Grosso, Damage detection and localization for continuous static monitoring of structures using a proper orthogonal decomposition of signals, Smart materials and structures, 15 ,2006.

[7] U. Galvanetto, G. Violaris, Numerical investigation of a new damage detection method based on proper orthogonal decomposition, Mechanical systems and signal processing, 1346-1361, 2007.

[8] C. Shane, R. Jha, Proper orthogonal decomposition based algorithm for detecting damage location and severity in composite beams, Mechanical systems and signal processing, 1062-1072, 2011.

[9] M.D. Buhmann, Radial basis functions: theory and implementations, Cambridge university press, 2003.

[10] V. Buljak, Inverse Analyses with Model Reduction: Proper Orthogonal Decomposition in Structural Mechanics, Springer, 2012.

[11] V. Buljak, G. Maier, Proper orthogonal decomposition and radial basis functions in material characterization based on instrumented indentation, Engineering Structures, 492-501, 2011.

[12] G. Bolzon, V. Buljak, An effective computational tool for parametric studies and identification problems in materials mechanics, Computational mechanics, 675-687, 2011.

[13] Chong, Edwin KP, and Stanislaw H. Zak. An introduction to optimization. Vol. 76. John Wiley & Sons, 2013.

[14] G. Venter, Review of optimization techniques, Encyclopedia of aerospace engineering, 2010.

[15] A. D. Belegundu and T. R. Chandrupatla, Optimization concepts and applications in engineering. New Jersey, CA: Prentice Hall, 1999.

[16] C.A. Rogers, A.J. Kassab, E.A. Divo, Z. Ostrowski, R.A. Bialecki, An inverse POD-RBF network approach to parameter estimation in mechanics, Inverse Problems in Science and Engineering, 20 (2012) 749-767.

[17] Z. Ostrowski, R. Białecki, A. Kassab, Solving inverse heat conduction problems using trained POD-RBF network inverse method, Inverse Problems in Science and Engineering, 39-54, 2008.

[18] A. Abraham, L. Jain, Evolutionary multiobjective optimization, Springer, 2005.

[19] M. Gen, R. Cheng, Genetic algorithms and engineering optimization, John Wiley & Sons, 2000.

[20] Kennedy, J., & Eberhart, R. Particle Swarm Optimization. IEEE International Conference on Neural Networks. Piscataway, NJ, USA. 1995.