ISSN 2176-5480

22nd International Congress of Mechanical Engineering (COBEM 2013)
November 3-7, 2013, Ribeirão Preto, SP, Brazil
Copyright © 2013 by ABCM

DETECTION OF HOLES IN A PLATE USING MULTIOBJECTIVE OPTIMIZATION AND MULTICRITERIA DECISION MAKING

Patricia da Silva Lopes Alexandrino
patty_lauer@unifei.edu.br
Ariosto Bretanha Jorge
ariosto.b.jorge@ unifei.edu.br
Sebastião Simões da Cunha Júnior
sebas@ unifei.edu.br
UNIFEI – Federal University of Itajubá, IEM – Institute of Mechanical Engineering
Av BPS, 1303, Itajubá, MG, Brazil – PO Box: 37500-903

Abstract. *In this work, an inverse problem of damage identification and localization in a structure is solved by a global optimization technique using genetic algorithms. The inverse problem is modeled as single and robust optimization problems. The damage is characterized by a hole in the structure, which modifies existing temperature and stress fields. The inverse problem is initially modeled as a mono-objective problem, where a functional formulation is minimized. After, the inverse problem is modeled as a robust optimization problem (multiobjective problem) where the optimum value and small variations around this optimum value are considered. Considering the multiobjective problem, two techniques are used to solve the damage detection problem. The first technique converts the Multiobjective Optimization Problem to a simple optimization problem through Weighting Objectives Method, and the second one uses a multiobjective genetic algorithm to obtain multiple solutions and a decision making method based on fuzzy theory to find the better tradeoff solution for the problem. Boundary element method is utilized to obtain the distribution of stress to elastostatic problem.*

Keywords: *Damage detection; multiobjective optimization; multicriteria decision making; genetic algorithm; boundary element method.*

1. INTRODUCTION

Several types of static and dynamic loads and the structural deterioration process can cause different types of structural damage. The damage can be characterized by a change in the structure, such as the presence of holes and cracks. The knowledge of the change in the material properties corresponding to the damage depends on the type of material and on the structural configuration. The proper assessment of the damage in a structure can be useful to infer its remaining service life.

The damage detection problem can be considered as a problem of system identification or an inverse problem. If considering an inverse problem of damage detection, this problem can be modeling through a direct problem, an inverse problem and uncertainty modeling. For the direct problem, a model is required to obtain information on the distribution of the quantity of interest throughout the structure, given the boundary conditions and the presence of the damage. For the inverse problem, a model is required for a procedure of locating damage in the structure given some (partial) information on the quantity of interest at some particular locations (for example, where some sensors are placed). Moreover, both direct and inverse problems are stochastic, therefore some kind of treatment of randomization needed to be performed at variables of the problems. Uncertainties are present in modeling of the plate structure under study, at damages in this plate structure and at numerical modeling of the problems.

The inverse problem of identifying the presence, location and size of damage, such as cracks and holes, in a plate structure can be modeled using optimization and parameter identification techniques. In Lopes *et al.* (2010), an inverse problem of identifying damage in a plate structure was solved using both optimization through genetic algorithm (GA), and parameter identification techniques through artificial neural network (ANN). These two independent techniques for inverse problem provide a means to verify the numerical results obtained for the location and size of the damage in the structure, increasing the confidence in the damage identification results.

Stochastic treatment can be performed through parameter identification procedure (for example, Kalman Filter, KF) or stochastic optimization (for example, or through Response Surface Methodology, RSM, or Monte Carlo Simulation, MCS). Procedures to obtain the response surface can include Design of Experiment (DoE) with regression, or the learning of structural behavior through a neural network procedure.

In stochastic optimization, robust optimization concept comes up. Optimal values to the objective functions and minimum variations of these functions at the optimal point vicinity are the goals of robust optimization. In this case, robust optimization is a Multiobjective Optimization Problem (MOOP). A MOOP can be solved through classic multiobjective optimization techniques and multiobjective evolutionary algorithms. The classic techniques (Weighting Objectives Method; Global Criterion Method) convert a MOOP to a simple optimization problem, i.e., a vector of objective functions is change to a single function that can be solved by some mono-objective optimization algorithm.

ISSN 2176-5480

Multiobjective evolutionary algorithm techniques can find multiple solutions (optimum solutions or non-dominated solutions) at the same time. Since multiple solutions are found, decision making methods (for example, a procedure based on fuzzy theory) can be used to find the better tradeoff solution for a given problem.

An explanation about multiobjective optimization using GA is present in (Konak *et al.*, 2006). In that work, formulation about multiobjective optimization, GA, and multiobjective GA are presented. In addition, some methodologies for multiobjective GA are described, shown advantage and disadvantage about each technique. Multiobjective optimization consists of two methodologies: *i*) to solve a problem as mono-objective optimization through matching all objectives in a single objective, or one objective is chose as the function to be optimized e the others objectives are moved to constraint set; *ii*) to find a optimal solutions set of Pareto (Pareto front or non-dominated solutions set). Some multiobjective evolutionary algorithms based on Pareto for this second methodology can be finding in Abido (2006) (NSGA – Nondominated Sorted Genetic Algorithm, NPGA – Niched Pareto Genetic Algorithm, and SPEA – Strength Pareto Evolutionary Algorithm). These algorithms are used in that work to solve a problem of non-linear multiobjective optimization of a power system. Furthermore, a decision making procedure based on fuzzy set theory to find the best tradeoff solution of Pareto front is presented.

Some authors have been developing works in damage detection area. Some works describe that damage in a metal is a process of initiation and growth of discontinuity in solid mean, such as, microcracks and voids (Lemaitre, 1984; Lemaitre and Dufailly, 1987). In continuum damage mechanics, material damage is a property that reduces material strength until failure. Damage can appear at some point of a geometric discontinuity, as an example, a hole or a crack. Thereby, tension distribution on a plate structure is not uniform at the section where there is a geometric discontinuity, that is, the tension value nearness of geometric discontinuity is greater than distant points on a structure. For simplification, a hole is considered as damage in this work.

The presence of damage may induce rapid changes in the field variable of the problem, and even discontinuities in the governing equation in the domain. Classical calculus-based optimization methods require evaluation of derivatives of the objective function, which may not be possible to be obtained, or may be numerically obtained, with unacceptable inaccuracy. Besides, these problems can have several local minima (multiple solutions), and thus a global optimization method (such as GA) is a better choice for the numerical solution (Stravoulakis and Antes, 1998; Engelhardt *et al.*, 2006).

Considering the direct problem modeling, numerical methods, such as BEM or the finite element method (FEM) can be used. BEM is used in (Martin *et al.*, 1995) where a new method was developed for finding deformations and tractions on parts of the boundary where these quantities are unavailable. This technique requires over-specified boundary on other parts of the boundary, i.e., both the displacements and the tractions must be specified at these other boundary subregions. The study or analysis of damage in a plate structure can be done through the distribution of stresses in this plate structure.

In this work, BEM approach in 2D was used for elastostatics problem. Damage is simulated by the presence of a hole inside the domain. Besides, the boundary conditions for the internal boundary of the plate structure (the hole) were set assuming zero traction. For each run of the direct model, the information about the location and radius of the hole, and also about the boundary conditions, loading, and plate structure and hole discretization, is also provided. After evaluating the boundary solution, the BEM code evaluates, as a post-processing, some quantities of interest at selected interior points. The selected interior points are candidates to be sensor locations, for a future experimental setting, and the quantities of interest at these points may be the quantities that these sensors are able to measure. Each run of the direct method using the elastostatics BEM formulation provides three pieces of information at an interior point – the components of the stress tensor, i.e., two normal stresses and one shear stress. The values of the normal stress and the shear stresses depend on the system of coordinates being used, or on the normal direction of the cutting plane that passes through the point of interest. As the goal of the inverse method is to identify and locate the hole, but not to identify any direction-dependent properties, the desired quantities to be supplied to the inverse model should be scalar quantities obtained at the selected interior points, and not direction-dependent quantities. Scalar quantities of interest can be obtained as the invariants of the stress tensor – in 2D, the mean stress and the octahedral stress – at the selected interior points. The mean stress and the octahedral stress are independent scalar fields, and either one can be used as the variable of interest at the selected interior points. In this work, the mean stress was adopted as the quantity to be provided to the inverse model for the elastostatics problem.

For the inverse problem, the direct BEM model first evaluates the differences in the quantity of interest (mean stress) between the undamaged plate and the plate with the damage, for all selected interior points. These differences are then supplied as input to the optimization (GA and Multiobjective GA) subroutines. The main idea for passing only differences of the quantities of interest is to avoid any possible bias related to the magnitude of these quantities, as only their change (due to the presence of the hole) is important for the inverse problem. The information provided by the BEM model for the direct problem is used for comparison with similar information, which must be available, for a plate with a hole with unknown size and location. Usually, the information on the “real” plate structure would be available by means of an experimental device, in which sensors would be put in all selected interior point locations. For the purpose of validating this approach, the plate structure with the “real” hole is also simulated with the BEM model, so the inverse problem algorithm will try to identify and locate this simulated “real” hole. Finally, in this work a robust optimization

ISSN 2176-5480

was performed, considering two functions: one related to the difference in mean stress between undamaged plate and the plate with the damage, other function related to minimum variations of this difference function at its optimal point vicinity. All subroutines in this work were written using the MATLAB® platform.

2. BOUNDARY ELEMENT METHODS FOR DIRECT PROBLEM

Numerical methods, such as BEM or FEM can be used for modeling the direct problem. In the FEM, the problem domain is partitioned into a number of subdomains (or finite elements) with connectivity between the elements provided through common nodal points. In the BEM, the governing partial differential equation of a domain is transformed into a set of integral equations, which relate the boundary variables (both known and unknown) (Basu, 2003; Brebbia and Dominguez, 1992). The BEM has some advantages with regard to FEM (Basu, 2003): *i*) BEM discretization is done only in the boundary of the domain, while FEM requires the discretization of the entire domain; *ii*) the number of equations associated with BEM is smaller than in the FEM approach, for the same degree of accuracy; *iii*) BEM is well suited for problems with singularities, such as in linear elastic fracture mechanics.

The BEM is a numerical procedure well adapted for the modeling of a structure with damage. In this method, the distribution of the quantities of interest in the domain is obtained from the information of the distribution of certain quantities in the boundary. Thus, the problem is described based on what happens in its boundaries, reducing the dimension of the problem and simplifying numerically the treatment.

In this work, the model investigated is the elastostatics formulation (see references (Brebbia and Dominguez, 1992; Paris and Cañas, 1997) for this formulation). A direct method for an elastostatics problem is modeled, where the distribution of the stresses on the external surface of a thin plate is analyzed. Without a hole, the distribution of the displacement and stresses is known a priori. If a small hole is included, this information is unknown and must be obtained numerically from the BEM solution. When modeling the damage detection problem by means of an analysis of the elastic response of the structure under excitation, perturbations in the expected response imply in the presence of damage. Thus, the damage in the structure will characterize its behavior, static or dynamic. In Lopes *et al.* (2010), the boundary integral equation formulation for elastostatic problem and the boundary element discretization were described.

3. OPTIMIZATION USING GENETIC ALGORITHMS

GA is a search method based on the processes of natural evolution. This method works with a set of possible solutions for a given problem, composing the initial population. In other words, GA uses multiple points to search for the solution rather than a single point in the traditional gradient based optimization method (Chou and Ghaboussi, 2001). In this algorithm the problem variables are represented as genes in a chromosome (individual). The chromosome for the damage detection problem can be assembled as showed in Lopes *et al.* (2010). Starting from an initial population, the individuals with better adapted genetic characteristics have higher chances of surviving and reproducing.

According to Burczynski and Beluch (2001), the GA's are methods that do not depend on the choice of the initial point, increasing the chances of obtaining the optimum global of the system. So that the population is diversified and maintain certain acquired adaptation characteristics by the previous generations, the genetic operators (selection, crossover and mutation) can be used. These operators transform the population through successive generations, extending the search until arriving to a satisfactory result. For more details about how these operators work, see references (Goldberg, 1998; Spall, 2003; Mitchell, 1999).

In this work, the optimal solution for unknown parameters of the damage (location and size) is obtained through the GA for elastostatics formulation of BEM. Considering this formulation (elastostatics formulation), the functional is defined as the difference between the measured (simulated) values of the local difference in the mean stress (between the undamaged plate and the plate with the damage) and the values of the same differences in mean stress calculated at the same points by the code (assuming several different locations and sizes for the “numerical” damage). The functional corresponds to the fitness function of the GA. The minimization of this fitness function allows the damage detection program to find the unknown parameters of the damage. The functional formulation is shown at Eq. (1).

$$J_j = \frac{1}{2}\sum_{i=1}^{n}\left(measured_i - calculated_{ji}\right)^2 \quad (1)$$

being n the number of internal points i (“sensors” placed in the plate) where the differences are evaluated; $measured_i$ the vector of simulated values for the differences obtained using BEM, for a given damage; and, $calculated_{ji}$ the vector of differences in mean stress (elastostatics formulation) calculated by the code for each individual j.

4. UNCERTAINTIES TREATMENT OF DAMAGE DETECTION PROBLEM

Considering uncertainties treatment, both direct and inverse problems are stochastic, therefore some kind of treatment of randomization needed to be performed at the variables of problems. Stochastic treatment can be performed through parameter identification procedures (KF) or stochastic optimization (RSM; MCS). At this work, stochastic optimization was used. DoE can be used to collect some data to stochastic optimization. DoE is a technique that creates a meta-model through regression function; therefore, a created response surface can be used at optimization algorithm. A reduction of experiments is possible using DoE, whereas it is possible to know influence of variables at performance of a given problem or process. Moreover, some improvement is obtained at results precision through detection of interaction among factors and detection of optimal levels of these factors (Montgomery and Runger, 2003).

When two or more objective functions need to be optimized in a problem, decision making procedures have to be accomplished in relation to multiobjectives. In other words, before solving some optimization problem, the programmer needs to decide how to treat the multiobjectives (for example, when two objective functions are presented, one function can be used as objective function and other one as constrained functions; or, both functions can be used as objective functions, combining and putting weight in the functions).

In stochastic optimization, robust optimization concept comes up. Optimal values to the objective functions and minimum variations of these functions at the optimal point vicinity are the goals of robust optimization. In this case, robust optimization also is a multiobjective problem and the optimal solutions for a problem are robust because these solutions are points in the feasible region where the values of objective function are insensible to small variations around these points.

4.1 Multiobjective optimization

Several problems have multiobjectives that need to be treated. These problems are known as Multiobjective Optimization Problems (MOOP) or Multiple Criterion Decision-Making (MCDM) problems. At MOOPs, the objective functions can be as maximized as minimized. Considering a minimization problem, the maximization functions need to be multiplied by (-1), change these functions to minimization functions. General mathematical formulation to the problem can be done according to Eq. (2) (Deb, 2001).

$$
\begin{aligned}
&\max_{x\in\Re^n} \text{ ou } \min_{x\in\Re^n} \left[f_1(x), f_2(x), \ldots, f_M(x) \right] \\
&s.t.\quad g_j(x) \geq 0, \quad j = 1, \ldots, J \\
&\qquad h_k(x) = 0, \quad k = 1, \ldots, K \\
&\qquad x_i^L \leq x_i \leq x_i^U, \quad i = 1, \ldots, n
\end{aligned}
\tag{2}
$$

with $f_i(\cdot): \Re^n \to \Re$, $g_j(\cdot): \Re^n \to \Re$ and $h_k(\cdot): \Re^n \to \Re$. Also, $x \in \Re^n$ is a vector of decision variables, x_i^L represents the inferior limit and x_i^U represents the superior limit of decision variables. The variable *n* represents a quantity of decision variables, *J* represents a quantity of inequality constraints and *K* represents a quantity of equality constraints.

A MOOP can be converted to a single optimization problem through classic techniques, such as Weighting Objectives Method and Global Criterion Method. When a single optimization problem is solved, only one solution is found in each run of the method. However, when some technique of multiobjective evolutionary algorithm is used to solve a MOOP, several solutions can be found at the same time. Then, a problem solved through these techniques has a set of optimal solutions, or Pareto-optimal solutions.

4.2 Pareto-optimal solutions

Pareto-optimal solution or Pareto front is a non-dominated solutions set (optimal solutions set) for a multiobjective problem. Evolutionary algorithms are used in MOOPs because a set of Pareto optimal solutions are found when these algorithms are run once. The first GA multiobjective is known as VEGA (Vector-Evaluated Genetic Algorithm) (Schaffer; 1984), however this algorithm does not present diversity mechanism, elitism, and each subpopulation is evaluated with relation to one different objective. At this algorithm, a population with *N* individuals is divided in *K* subpopulations with equal size. VEGA has a disadvantage to converge to extreme of each objective (Konak *et al.*, 2006). Others evolutionary algorithms appear after VEGA, such as MOGA - Multiobjective Genetic Algorithm (Fonseca and Fleming, 1993), NPGA - Niched Pareto Genetic Algorithm (Horn, 1994), NSGA - Non-Dominated Sorting Genetic Algorithm (Srinivas and Deb, 1994), NSGA-II - Fast Non-Dominated Sorting Genetic Algorithm (Deb *et al.*, 2000 and 2002), SPEA - Strength Pareto Evolutionary Algorithm (Zitzler and Thiele, 1999), SPEA-2 (Zitzler *et al.*, 2001), PESA-II - Region-based Selection in Evolutionary Multiobjective Optimization (Corne, 2001), and so on.

These algorithms are different each one at some characteristic (manner of assigning value of fitness function to individuals; elitism, or diversification mechanism).

The first evolutionary algorithm used to solve a MOOP was NSGA. This algorithm is different from simple GA only in manner that selection operator works. However, NSGA presents high computational complexity, lack of elitism, and need of specifying a sharing parameter. Then, NSGA was changed and a new version appears with name NSGA-II (Deb *et al.*, 2000). NSGA-II is a multiobjective algorithm based in GA that sorts the solutions using Nondominated Sorting technique. In this technique, a population is divided in groups of individuals (or fronts) according to dominance level. In each subset, no individual or solution dominates anybody. The best solutions to current generation are the individuals of first front therefore, on the first front are present the individuals that are not dominated by no other individual of generation. So, these individuals are near to the Pareto line and they are known as non-dominated solutions. Besides, NSGA-II uses an operator known as crowding distance to estimate density of solutions around a point or individual. This operator allows a uniform spread of solutions along the Pareto lines. The crowding distance operator sorts each individual according to its distance in relation to neighbor points on the same front (in relation to each objective) (Deb *et al.*, 2000).

At MATLAB® (version R2008a or superior), the function *gamultiobj* returns a Pareto front using GA. This function is a variation of NSGA-II algorithm where GA uses a controlled elitism. A controlled elitism enables that the population diversity increase even so an individual has low value of fitness. Elitism is controlled through two parameters: *ParetoFraction* that limits the number of individuals of population on Pareto front, and *DistanceFcn* that avoids individuals that are far from this front. Besides, a function can be written to calculate distance measure of individuals (measure of the concentration of the population), or the *distancecrowding* function can be used as default. The *distancecrowding* function presents an extra parameter that computes the distance in decision variable or design space (genotype) or in function space (phenotype).

4.3 Decision Making based on fuzzy set theory

Decision making methods can determine the best tradeoff solution for the problem before some multiobjective evolutionary algorithm finding the non-dominated solutions set (Pareto front). All solutions on Pareto front are equivalent (with the same dominance level). Then, a procedure based on fuzzy set theory can be used to accomplish the decision work.

Traditional logic works only with true or false (exact values), on the other hand, fuzzy logic (fuzzy set) works with “degrees of truth” or “degrees of false” (imprecise information). Expressions like “more or less” and “maybe” can be mapped with fuzzy logic. In fuzzy logic, due to the imprecise nature of the decision maker’s judgment, the *i*-th objective function f_i of a solution on Pareto front is represented by a membership function μ_i (Abido, 2006). The values of membership function designate the level of achievement of the objective functions of some problem, and these values are between 0 and 1. There are several kinds of membership functions, such as, linear, triangular, trapezoidal, or exponential membership functions (Sakawa *et al.*, 1987). In this work, a trapezoidal membership function was used (Eq. (3)).

$$\mu_i = \begin{cases} 1; & f_i \le f_i^{\min} \\ 0; & f_i \ge f_i^{\max} \\ \dfrac{f_i^{\max} - f_i}{f_i^{\max} - f_i^{\min}}; & f_i^{\min} < f_i < f_i^{\max} \end{cases} \tag{3}$$

where $f_i^{\min}$ and $f_i^{\max}$ are the minimum and maximum values of the *i*-th objective function, respectively.

For each nondominated solution k, and regarding N_{dom} the number of nondominated solution of the Pareto front, the membership function is normalized according to Eq. (4).

$$\mu^k = \frac{\sum_{i=1}^{N_{obj}} \mu_i^k}{\sum_{k=1}^{N_{dom}} \sum_{i=1}^{N_{obj}} \mu_i^k} \tag{4}$$

The Best tradeoff solution is find through the maximum value of μ^k, considering Eq. (5).

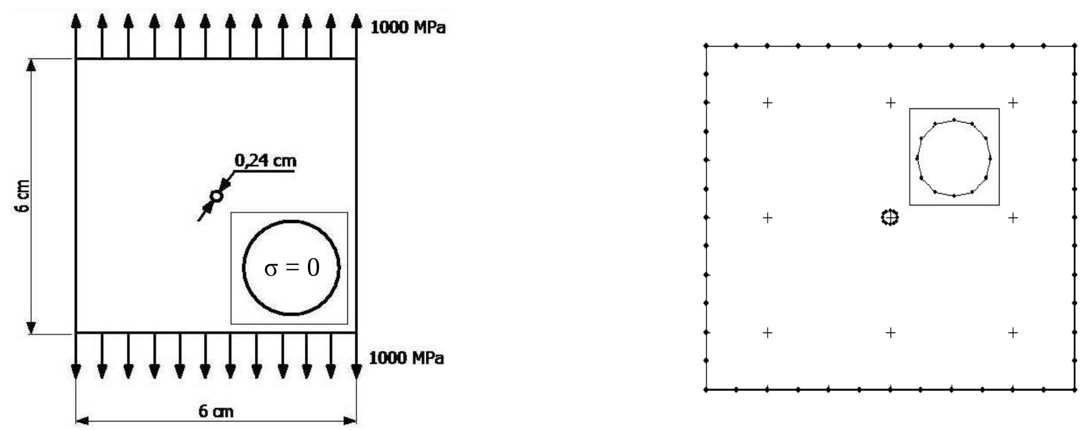
$$\gamma = m\acute{a}x(\mu_k); \quad k = 1, 2, \ldots, N_{dom} \tag{5}$$

The values of $f_i^{\min}$ and $f_i^{\max}$ can be find after ranking the nondominated solutions on Pareto front. The worst solution for ***i***-th objective solution is the variable $f_i^{\max}$.

In fuzzy set there are terms known as qualifiers or modifiers that modify the shape of fuzzy sets. The qualifiers allow mapping in fuzzy values the human language that is full of imprecise terms. The qualifiers "very", "more important", "few", and "more or less" are some examples of these terms. The fuzzy qualifiers are similar to weight of a function and they are used at the membership function.

5. NUMERICAL RESULTS AND DISCUSSION

For the elastostatics problem, a BEM model was built for the plate with a hole with the boundary conditions illustrated in Fig. 1(a). A mesh with 48 constant elements was implemented for the external contour (outer boundary) and 12 constant elements in the hole, nine sensors were considered on the plate surface (Fig. 1(b)). At the present work, the sensors were uniformly distributed on the plate and no positioning study of the sensors was performed. The plate was simulated with shear modulus equal to 94,500 MPa and a Poisson's ratio for plane strain equal to 0.1.



(a) (b)

Figure 1. Plate model: (a) dimensions, loading, and boundary conditions. Insert shows a stress-free hole; (b) boundary discretization and sensor locations. Insert shows hole discretization.

5.1 Formulation of multiobjective optimization problem

The problem of damage detection in a thin plate can be formularized as a mono-objective optimization problem or as a multiobjective optimization problem, both of them using GA. The mono-objective optimization problem was presented and solved in Lopes *et al.* (2010). So, in this present work, the multiobjective optimization problem is considered.

For solved the multiobjective optimization problem as a robust optimization problem, initially a function for the standard deviation of the functional formulation (Eq. (1)) needs to be found. This function corresponds to the square root of variance of the functional. The variance function is obtained through a multivariate regression with terms until third order. The independent variables are information about holes (*x*- and *y*-coordinates of its center, and also its radius) and the dependent variable is the standard deviation of the functional formulation for each hole. A matrix with three columns (*x*, *y* and radius) and 275 lines was formed. This matrix was assembled considering that the holes had radius from 0.10 cm to 0.15 cm, with step size of 0.005 cm, and the center of holes were considered at positions from 1 cm to 5 cm, with step size of 1 cm for coordinates *x* and *y*.

In a first analysis, the elements of the matrix were considered as the mean values (μ) of a uniform distribution (the analysis was also done for the mean values of a normal distribution). The standard deviation for each hole localization (σ) was considered equal to 10% of one-third of a unit ($3\sigma = 0.10$), and the standard deviation for each radius value (σ_r) was considered equal to 10% of one-third of the mean value for the radius value ($\sigma_r = 0.10\mu_r/3$, or, $3\sigma_r = 0.10\mu_r$). Then, a region near to the mean values (of parameters *x*, *y* and *r*) was found, where 100 random numbers from a uniform distribution were placed (another analysis for normal distribution was also done). This region was found through two limits, an inferior (inf) and a superior (sup) limit, near the mean value. The inf limit was guessed as the mean value minus 10% of amplitude value and the sup limit was guessed as the mean value plus 10% of amplitude value (Eq. (6)).

ISSN 2176-5480

$$\begin{aligned} \inf &= \mu - 0.10 amplitude = \mu - 0.60\sigma \\ \sup &= \mu + 0.10 amplitude = \mu + 0.60\sigma \end{aligned} \quad (6)$$

The amplitude value (interval of values) is the difference between the mean value plus three times the standard deviation and the mean value minus three times the standard deviation, according to Eq. (7).

$$amplitude = (\mu + 3\sigma) - (\mu - 3\sigma) = 6\sigma \quad (7)$$

These inf and sup limits were used to find random numbers, as shown in Eq. (8), for each *x*, *y* and *r* variables.

$$\text{variable} = inf + (sup - inf)\,\text{rand}(100,1) \quad (8)$$

In this Eq. (8), rand(100,1) is a function that uniformly distributed 100 (a hundred) pseudorandom numbers on an interval between 0 and 1. When the function randn(100,1) is used, this function returns 100 (a hundred) pseudorandom numbers drawn from a normal distribution with mean 0 and standard deviation 1.

Now, with the “measured” holes (determined by mean values of its parameters; the “measured” hole is the “real” hole on the plate) and the “calculated” holes distributed around each “measured” hole, the mean stress was found for each hole through a routine that use the BEM. These mean stress values were found on 9 interior points of plate structure (sensors at positions (1.0;1.0) cm, (1.0;3.0) cm, (1.0;5.0) cm, (3.0;1.0) cm, (3.0;3.0) cm, (3.0;5.0) cm, (5.0;1.0) cm, (5.0;3.0) cm, and (5.0;5.0) cm). Then, the “measured” and the “calculated” values of mean stress were used by the functional formulation (Eq. (1)). With this technique, the functional variance could be found for each hole.

The last step was to find a functional variance function for each hole. A natural logarithm of values was used for a change of scale, then, a multivariate regression was performed with regard to x, y and r parameters (considering a 95% confidence level). Since the place of sensors was not considerate at computations of the functional variance function, discontinuities can be avoided at this function. The multivariate regression presented a R^2 value equal to 72.8% and a p-value equal to 0 for the uniform distribution, and a R^2 value equal to 83.9% and a p-value equal to 0 for the normal distribution. As the regression function for normal distribution presented a greater R^2 value than the function for uniform distribution, that regression function was chosen as the functional variance function.

The regression function for normal distribution is presented in Eq. (9). In this function, all regression terms were considered.

$$\begin{aligned} J_{\text{var}} = &-1.553\times10^{-5} + 9.857\times10^{1} r - 1.829x - 1.070y - 9.339\times10^{2} r^2 + 2.610\times10^{-1} x^2 + 6.083\times10^{-2} y^2 + \\ &+ 8.038rx + 3.733ry + 2.421\times10^{-1} xy + 2.696\times10^{3} r^3 - 2.503\times10^{-2} x^3 + 1.703\times10^{-2} y^3 - 1.296\times10^{1} r^2 x + \\ &- 2.361\times10^{1} r^2 y - 6.689\times10^{-1} x^2 r + 4.175\times10^{-2} x^2 y + 3.838\times10^{-1} y^2 r - 7.355\times10^{-2} y^2 x - 1.458\times10^{-1} x\,y\,r \end{aligned} \quad (9)$$

5.2 Analysis of the results from multiobjective optimization problem

The objectives (minimum value of functional J and minimum value of its variance) of damage detection problem were solved through a variation of NSGA-II algorithm. The initial population was assembled with 168 individuals, considering the presence of sensors information (difference at mean stress for elastostatic formulation of BEM). The holes of this population had radius equal to 0.10 cm, 0.125 cm and 0.15 cm, and the *x* and *y*-coordinate of the center of the hole was varied from 1.0 cm to 5.0 cm with a step size of 0.5 cm. The place of sensors was not considerate in the initial population.

The Pareto front for a hole in (1.0;2.0) cm and radius equal to 0.12 cm is showed in Fig. 2. This Pareto front was obtained in generation equal to 106.

ISSN 2176-5480

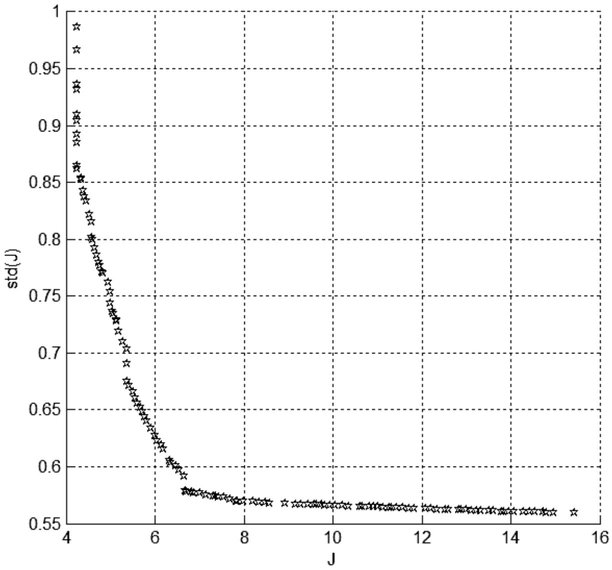


Figure 2. Pareto front obtained by multiobjective GA.

The number of generations equal to 200 and a tolerance less than or equal to 1×10^{-6} were assumed as stopping criteria. The individuals (parents) for next generation were selected through a tournament and the heuristic function (with a value of ratio equal to 0.9) was assumed as crossover function. The mutation function adopted was the *adaptive feasible* function (Matlab®, version R2008a) that randomly generates directions that are adaptive with respect to the last successful or unsuccessful generation. The crossover fraction was set equal to 0.95 (95%) and the mutation fraction was considered equal to 0.05 (5%). The migration of individuals was considered in both directions and the elitism was considered equal to 1 individual. The Pareto fraction was considered equal to 0.75, that is, 75% of individuals are kept on the first Pareto front while individuals from higher fronts are selected. The *distancecrowding* function ((Matlab®, version R2008a)) (that computes distance measure of an individual), computed in function space (phenotype), was used.

The number of points on the Pareto front (Fig. 2) was equal to 126. The value of the membership function (Eq. (3)) was found for each non-dominated solution on the Pareto front and this values set was normalized (Eq. (4)). Then, the best tradeoff solution was found through Eq. (5) to the values set of normalized membership function. Finally, the hole location (x, y) and radius r of this hole can be found through the best tradeoff solution.

Figure 3 shows the representation of 126 points of Pareto front is non continuous (dashed) line, the "real" hole is represented in continuous line, and the results ("Result 1", "Result 2", and "Result 3") from fuzzy decision making (considering different fuzzy qualifiers) is represented in dash-dot line. The "real" hole, Result 1, and some results from multiobjective algorithm are showed with more details at zoom area in this Fig. 3.

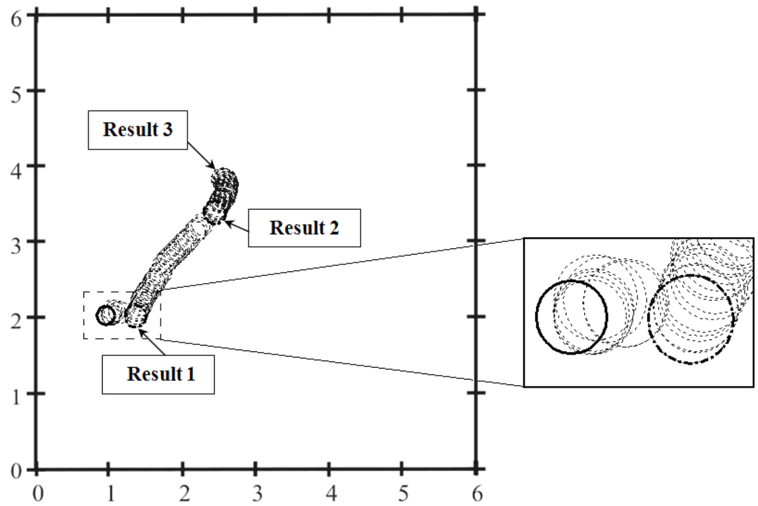


Figure 3. "Real" hole (full line), holes found by the variation of NSGA-II algorithm (dashed line), and results from fuzzy decision making (dash-dot line).

ISSN 2176-5480

In this Fig. 3, "Result 1" is the result from fuzzy decision making where the functional formulation function J is "more important" than standard deviation ($\sqrt{J_{\text{var}}}$) of this functional. In other words, the weight of J can be equal to $0.8\mu_J$ (then, the weight of $\sqrt{J_{\text{var}}}$ can be $(1-0.8)\mu_{std(J)}$), or the weight of J can be equal to $0.9\mu_J$ (then, the weight of $\sqrt{J_{\text{var}}}$ can be $(1-0.9)\mu_{std(J)}$). Therefore, a hole was found at location $x=1.40$ cm, $y=1.99$ cm, and radius $r=0.15$ cm. "Result 2" presents the result for no importance (without to use the fuzzy qualifier) to the functions (functional formulation and its standard deviation). "Result 3" is the result for the case where only the standard deviation of functional formulation is present ($0\mu_J$ e $1\mu_{std(J)}$). This last result ("Result 3") corresponds to the hole more distant from "real" hole. Finally, the deterministic result ($1\mu_J$ e $0\mu_{std(J)}$) correspond to the hole closer to the "real" hole and the location of that hole is $x=1.06$ cm, $y=2.00$ cm, and radius $r=0.12$ cm.

Now, considering an initial population with only 6 individuals (1 1 0.10; 1 2 0.14; 1 4 0.12; 3 2 0.11; 3 4 0.13; 5 5 0.10) without the presence of sensors information (difference at mean stress for elastostatic formulation of BEM) in this population, damage detection problem were solved. The BEM routine (considering sensor information) was used when the fitness function was available.

The number of generations equal to 15 and a tolerance less than or equal to 1×10^{-6} were assumed as stopping criteria. The crossover fraction was set equal to 0.75 (75%) and the mutation fraction was considered equal to 0.25 (25%). No elitism was considered for this simulation. The others parameters were set according to the case where the sensors information was present at GA population.

The results found by multiobjective GA to a hole at (1.0,2.0) cm and radius equal to 0.12 cm are showed in Tab. 1. In this Tab. 1, the values of J and $\sqrt{J_{\text{var}}}$ correspond to the non-dominated solutions of problem. These values (J and $\sqrt{J_{\text{var}}}$) are used to find the membership values (Eq. (3)), that are normalized (Eq. (4)), resulting at the values μ^k presented in sixth column of Tab. 1.

Then, the best tradeoff solution is equal to 0.328 (Eq. (5)). This value is present in second and fourth line of Tab. 1. After finding the best tradeoff solution, a hole at $x=1.00$ cm, $y=2.00$ cm, and radius $r=0.11$ cm can be determined.

Table 1. Results for multiobjective GA using fuzzy decision making without the sensors information in the GA population.

J MPa	$\sqrt{J_{\text{var}}}$ MPa	x cm	y cm	r cm	μ^k
19.467	0.640	2.969	4.000	0.130	0.050
7.002	1.048	1.000	2.000	0.109	0.328
10.654	0.938	1.031	2.012	0.138	0.245
7.002	1.048	1.000	2.000	0.109	0.328
19.467	0.640	2.969	4.000	0.130	0.050

The results of Tab. 1 are presented in Fig. 4. The number of points on the Pareto front was equal to 5. The result obtained from multiobjective GA approach using fuzzy decision making shows that the exact location of "measured" hole was found.

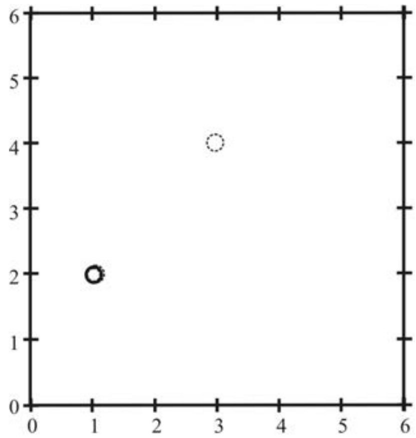


Figure 4. Graphical representation of 5 points from Pareto front (dashed line) and "real" hole (full line).

6. CONCLUSIONS

In this work, an inverse problem of identifying damage in a plate structure was solved by a global optimization technique using genetic algorithms. The inverse problem was modeled as a multiobjective problem where minimum value of functional formulation and minimum variations around this optimum value are considered. Then, the multiobjective problem was modeled as a robust optimization problem where the results are not sensible to small variations around the optimal points. For the direct model in the inverse problem, an elastostatics problem was modeled through a formulation of boundary element method (BEM). The analysis of the results indicates that the damage detection code using GA find a region for the probable occurrence of the hole. In other words, the region is found due to own randomness of the GA approach that generates a different optimal solution every time it is run. Considering the use of different techniques to obtain the results for multiobjective problem, and the use of different populations of GA, a comparison between these results could be done. Besides, the computational cost to solve the problem, when the sensors information is not presented in the GA population, was bigger than the program where this information was present in the population. This difference at computational cost occurred because the BEM routine was executed several times during the run of the damage detection.

7. ACKNOWLEDGEMENTS

The authors would like to acknowledge the financial support from the Brazilian agencies CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico, and FAPEMIG – Fundação de Amparo à Pesquisa do Estado de Minas Gerais, and also the financial support from AFOSR - Air Force Office of Scientific Research.

8. REFERENCES

Abido, M. A. (2006); “Multiobjective Evolutionary Algorithms for Electric power Dispatch Problem”; *IEEE Transactions on Evolutionary Computation*, v. 10, num. 3, pp. 315-329.

Basu, P.K.; Jorge, A.B.; Badri, S. and Lin, J. “Higher-Order Modeling of Continua by Finite-Element, Boundary-Element, Meshless, and Wavelet Methods”, *Computers and Mathematics with Applications*. 46 (2003), pp. 15-33.

Brebbia, C.A. and Dominguez, J. *Boundary Elements: An Introductory Course*, 2nd ed., McGraw-Hill, Boston, 1992.

Burczynski, T. and Beluch, W. “The identification of crack using boundary elements and evolutionary algorithms”, *Engineering Analysis with Boundary Elements*. 25 (2001), pp. 313-322.

Chou, J.H. and Ghaboussi, J. “Genetic algorithm in structural damage detection”, *Computers and Structures*. 79 (2001), pp. 1335-1353.

Corne, D.; Jerram, N. R.; Knowles, J.; Oates, J. (2001); “PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization”. In: *Proceeding of the Genetic and Evolutionary Computation Conference (GECCO - 2001)*, San Francisco, CA.

Deb, K.; Agrawal, S.; Pratab, A.; Meyarivan, T. (2000); “A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II”, *Lecture Notes in Computer Science*, 2000, v. 1917, pp. 849-858.

Deb, K. (2001), *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc., New York, NY.

Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. (2002); “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”. *IEEE Transactions on Evolutionary Computation*; v. 6, num. 2, pp. 182-197.

Engelhardt, M.; Stavroulakis, G.E. and Antes, H. “Crack and flaw identification in elastodynamics using Kalman filter techniques”, *Computational Mechanics*. 37 (2006), pp. 249-265.

Fonseca, C. M.; Fleming, P. J. (1993); “Multiobjective Genetic Algorithms”. In: *IEE Colloquium on ‘Genetic Algorithms for Controls Systems Engeneering* (Digest No. 1993/130), 28 May 1993. London, UK: IEE.

Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Co, Massachusetts, 1998.

Horn, J.; Nafpliots, N.; Goldberg, D. E. (1994); “A Niched Pareto Genetic Algorithm for Multiobjective Optimization”. In: *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computional Inteligence*, 27-29 June 1994. Orlando, FL, USA: IEEE.

Konak, A.; Coit, D.W. and Smith, A.E. “Multi-objective optimization using genetic algorithms: A tutorial”, *Reliability Engineering & System Safety*. v. 91 (2006), pp. 992-1007.

Lemaitre, J. “How to Use Damage Mechanics”, *Nuclear Engineering and Design*. 80 (1984), pp. 233-245.

Lemaitre, J. and Dufailly, J. “Damage Measurements”, *Engineering Fracture Mechanics*. 28 (1987), No.5/6, pp. 643-661.

Lopes, P.S.; Jorge, A.B.; Cunha Jr., S.S. “Detection of holes in a plate using global optimization and parameter identification techniques”. *Inverse Problems in Science & Engineering. IPSE (2010)*, v. 18(4), pp. 439-463.

Martin, T.J.; Halderman, J.D. and Dulikravich, G.S. “An inverse method for finding unknown surface tractions and deformations in elastostatics”, *Computers & Structures*. 56, No. 5. (1995), pp. 825-835.

ISSN 2176-5480

Mitchell, M. *An Introduction to Genetic Algorithms*, 5th Edition, MIT Press, Cambridge, Massachusetts-London, England, 1999.

Montgomery, D. C.; Runger, G. C. (2003); *Estatística aplicada e probabilidade para engenheiros*, 2º Edição, LTC – Livros Técnicos e Científicos Editora S.A.

Paris, F. and Cañas, J. *Boundary Element Method - Fundamentals & Applications*, Oxford Univ. Press, New York, 1997.

Portela, A.; Aliabadi, M.H. and Rooke, D.P. "The dual boundary element method: Effective implementation for crack problems", *International Journal for Numerical Methods in Engineering*. 33 (1992), pp. 1269-1287.

Schaffer, J. D. (1984); *Some experiments in machine learning using vector evaluated genetic algoritms*. Ph. D. Thesis in Electrical Engineering, Vanderbilt University, Nashville.

Spall, J.C. *Evolutionary Computation I: Genetic Algorithms*, in *Introduction to stochastic search and optimization: estimation, simulation, and control*, John Wiley & Sons, Inc., 2003

Srinivas, N.; Deb, K. (1994); "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms". *Evolutionary Computation*, v. 2, num. 3, pp. 221-248.

Stravoulakis, G.E. and Antes, H. "Flaw identification in elastomechanics: BEM simulation with local and genetic optimization", *Structural Optimization*, Springer-Verlag. 16 (1998), pp. 162-175.

Zitzler, E.; Thiele, L. (1999); "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strenght Pareto Approach", *IEEE Trans Evol Comput 1999*; v. 3, num. 4, pp. 257-71.

Zitzler, E.; Laumanns, M.; Thiele, L. (2001); "SPEA2: Improving the Strenght Pareto Evolutionary Algorithm". Swiss Federal Institute Technology: Zurich, Switzerland.

9. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.