SDSS'Rio 2010 STABILITY AND DUCTILITY OF STEEL STRUCTURES
E. Batista, P. Vellasco, L. de Lima (Eds.)
Rio de Janeiro, Brazil, September 8 - 10, 2010

STRUCTURAL DAMAGE ASSESSMENT USING THE DIFFERENTIAL EVOLUTION AND THE ANT COLONY OPTIMIZATION TECHNIQUES

Genasil F. dos Santos[1], José Guilherme S. da Silva[2], Francisco J. da C. P. Soeiro[2]

[1] State University of Rio de Janeiro, UERJ
Civil Engineering Post-Graduate Programme, PGECIV
e-mail: genasil@yahoo.com.br

[2] State University of Rio de Janeiro, UERJ
Mechanical Engineering Department, MECAN
e-mail: jgss@uerj.br, soeiro@uerj.br

Keywords: structural damage assessment, system identification techniques, inverse problems, differential evolution optimization technique, ant colony optimization technique.

Abstract. *Structural systems in a variety of applications including aerospace vehicles, automobiles and engineering structures such as tall buildings, bridges and offshore platforms, accumulate damage during their service life. The approach used in this investigation is one where the structural properties of the analytical model are varied to minimize the difference between the analytically predicted and empirically measured response. This is an inverse problem where the structural parameters are identified. In this work a reduced number of vibration modes and nodal displacements were used as the measured response. For the damage assessment problem a finite element model of the structural system is available and the model of the damaged structure will be identified. Damage will be represented by a reduction in the elastic stiffness properties of the system. In this investigation, the Differential Evolution (DE) and the Ant Colony Optimization (ACO) were applied to simple truss structures with different levels of damage.*

1 INTRODUCTION

In a typical load bearing structure, degradation of structural properties due to damage manifests itself as a change in the static and dynamic response. A correlation of the measured response with that obtained from an analytical model of the undamaged structure, allows for the possibility of determining a modified model that predicts the altered response. This inverse problem is solved using a system identification technique [1].

In this paper the output error approach of system identification is used to determine changes in the structural parameters that result from structural damage. Damage is represented by reduction in the elastic properties of the element. The net changes in these quantities due to damage are lumped into a single coefficient d_i for each element that is used to multiply the stiffness matrix of that particular element. These coefficients d_i constitute the design variables for the resulting optimization problem.

Static displacements and eigenmodes are used as measured data for the inverse problem of damage detection. Reduced sets of eigenmodes and static displacements are used [2-3]. The approach of considering one design variable d_i for each element in the structure usually results in a large dimensionality problem.

These results in a very nonconvex design space, probably with several local minima, where Gradient-based nonlinear methods for function minimization may have difficulties to find the global optimum. In this work two global optimization methods, the Differential Evolution and the Ant Colony Optimization which are two heuristic population based methods were used for function minimization.

2 STRUCTURAL DAMAGE ASSESSMENT

In a finite element formulation, structural characteristics are defined in terms of the stiffness, damping, and mass matrices [K], [C] and [M], respectively. The governing equation of equilibrium for a dynamical system involves each of these matrices, and can be written using Equation (1). In Equation (1), x is the displacement vector and P(t) is the vector of applied loads. The static load-deflection relation only involves the system stiffness matrix, as presented in Equation (2).

$$[M]\ddot{x} + [C]\dot{x} + [K]x = P(t) \tag{1}$$

$$[K]x = P \tag{2}$$

The analytical model describing the eigenvalue problem for an undamped system can be stated in terms of the system matrices defined above, the i-th eigenvalue ω_i^2, and the corresponding eigenmode Y_i as follows:

$$([K] - \omega_i^2 [M])Y_i = 0 \tag{3}$$

It is clear from these equations that a change in the system matrices results in a different structural system response and this difference can be related to changes in specific elements of the system matrices. Since internal structural damage typically does not result in a loss of material, it will assume the mass matrix to be constant.

The stiffness matrix can be expressed as a function of the thickness (t), the length (L), the cross-sectional area (A), the Young’s modulus (E), and the flexural and torsional stiffness (EI) and (GJ), respectively. The stiffness matrix of the truss element modified to include the damage coefficient is given by Equation (5), where C = cos α and S = sin α. The truss element is shown in Figure 1. The numerical approach was applied to simple steel truss structures, see Figure 2, with different levels of damage.

$$[K] = [K(t, L, A, E, EI, GJ)] \tag{4}$$

In this paper, changes in these quantities are lumped into a damage coefficient d_i, which is used to multiply the stiffness matrix of a particular element. The coefficients d_i constitute the design variables for the damage assessment problem and vary from 0 (undamaged element) to 1 (completely damaged element). The values of the coefficients d_i give the location and the extent of damage in the structure.

$$K_i^{(.)} = \frac{(1-d_i)E_i A_i}{L_i} \begin{bmatrix} C^2 & CS & -C^2 & -CS \\ CS & S^2 & -CS & -S^2 \\ -C^2 & -CS & C^2 & CS \\ -CS & -S^2 & CS & S^2 \end{bmatrix} \tag{5}$$

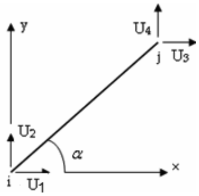
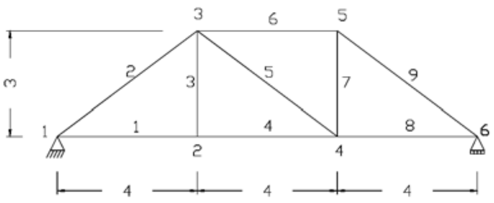
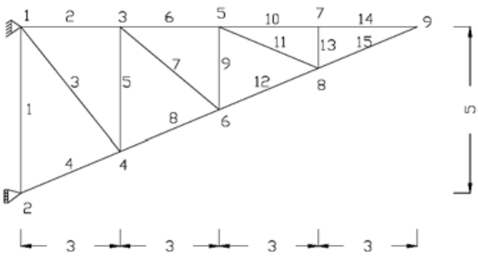


Figure 1: Truss element.





a) Nine bar truss. b) Fifteen bar truss.

Figure 2: Investigated steel truss structures (dimensions in meters).

If the measured and analytically determined static displacements or vibration modes are denoted by Y_m and Y_a, respectively, the optimization problem can be formulated as determining the vector of design variables d_i that minimize the scalar objective function representing the difference between the analytical and experimental response, as presented in Equation (6), where i represents the degree of freedom and j denotes a static loading condition or a particular vibration mode.

$$F = \sum_{ij} \left\| Y_m^{ij} - Y_a^{ij} \right\|^2 \tag{6}$$

One important advantage of this approach is that the complete set of modes or displacements is not needed since the objective function involves only the difference between components of those vectors. Some of the components may be neglected according to its importance in the behaviour of the structure. In this paper, for damage assessment purposes, only the vertical displacements were used. They are dominant and easier to measure. The other components are relatively small and neglecting them in the objective function does not affect the process of damage detection. The approach still works and becomes more realistic since in large structures only few dominant displacements can be obtained accurately. Also in the cases where eigenmodes were used for damage assessment purpose, only the first four modes and the respective eigenvalues (natural frequencies) were used in the objective function.

The objective function presented in Equation (6) was minimized with two global optimization techniques: Differential Evolution (DE) and the Ant Colony Optimization (ACO). These methods are described briefly in the next section.

3 OPTIMIZATION TECHNIQUES

3.1 Differential Evolution (DE)

The Differential Evolution (DE) was proposed by Storn and Price [4] as an algorithm to solve global optimization problems of continuous variables. The main idea behind DE is how possible solutions taken from the population of individuals are set, recombined and chosen to evolve the population to the next generation. In a population of individuals, a fixed number of vectors are randomly initialized, and then evolved over the optimization task to explore the design space and hopefully to locate the optimum of the objective function. At each iteration, new vectors are generated by the combination of vectors randomly chosen from the current population.

This operation is called “mutation” and a mutant population is created. The outcoming vectors are then mixed with a predetermined target vector. This operation is called “crossover” or “recombination” and produces a “trial vector”. Finally, the “trial vector” is accepted for the next generation if it yields a reduction in the value of the objective function. This last operation is referred to as “selection”.

As can be seen, the basic algorithm preserves some common aspects of the traditional simple Genetic Algorithm (GA), specially the nomenclature of selection, crossover and mutation. A population of individuals can be expressed as a matrix given by Equation (7), where *i* is the number of individuals of the population and *j* is the number of design variables.

$$P = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1j} \\ x_{12} & x_{22} & \dots & x_{2j} \\ \dots & \dots & \dots & \dots \\ x_{i1} & x_{i2} & \dots & x_{ij} \end{bmatrix} \tag{7}$$

As described before, the "mutation" operator adds the weighted difference between two individuals to a third individual (base vector). There are several ways to mutate a variable and the equation below shows a possible mutation scheme, among others.

$$v_i = x_{r1} + F\left(x_{r2} - x_{r3}\right) \tag{8}$$

In Equation (8), $\mathbf{v_i}$ is the mutant vector, x_{r1}, x_{r2} and x_{r3} are random integer indexes and mutually different, F is a real constant factor which controls the amplification of the differential variation and x_{best} is the best individual of the current population. The next operation is "crossover". Each mutant vector is combined with a target vector $\mathbf{x_i}$. This operation is performed by swapping the contents of the mutant vector with the correspondent component of the target vector based on a crossover probability (CR). The resulting vector is denominated "trial vector".

At the sequence of the DE algorithm, the selection operator decides whether or not the new vector $\mathbf{x_{trial}}$ should become a member of the next generation. This is decided by the objective function value of all new individuals f($\mathbf{x_{trial}}$) which are compared with the one of the target vector f($\mathbf{x_{target}}$). If there is an improvement, $\mathbf{x_{trial}}$ is selected to be part of the next generation, otherwise, $\mathbf{x_{target}}$ is kept. According to Storn and Price [5] it is recommended that the population has a size of 10 times the number of design variables, the crossover probability, CR, usually is chosen in the range [0,1] and the weight factor F is usually chosen in the interval [0,2]. In this paper CR was set as 0.8 and F was set as 0.9.

3.2 Ant Colony Optimization (ACO)

The Ant Colony Optimization method was proposed by Dorigo [6]. The native ants are capable of finding the shortest path from a food source to the nest without using visual cues. Also, they are capable of adapting to changes in the environment, for example finding a new shortest path once the old one is no longer feasible due to a new obstacle.

It is well-known that the main means used by ants to form and maintain the line is a pheromone trail. Ants deposit a certain amount of pheromone while walking, and each ant probabilistically prefers to follow a direction rich in pheromone rather than a poorer one. This elementary behaviour of real ants can be used to explain how they can find the shortest path which reconnects a broken line after the sudden appearance of an unexpected obstacle has interrupted the initial path.

In fact, once the obstacle has appeared, those ants which are just in front of the obstacle cannot continue to follow the pheromone trail and therefore they have to choose between turning right or left. In this situation one can expect half the ants to choose to turn right and the other half to turn left.

It is interesting to note that those ants which choose, by chance, the shorter path around the obstacle will more rapidly reconstitute the interrupted pheromone trail compared to those which choose the longer path. Hence, the shorter path will receive a higher amount of pheromone in the time unit and this will in turn cause a higher number of ants to choose the shorter path. Due to this positive feedback process, very soon all the ants will choose the shorter path.

The ACO was developed initially for combinatorial optimization only. Particularly good results were obtained in the solution of the Problem of the Traveling Salesman [7]. The damage assessment problem deals with continuous variables. In this work an extension of the ACO algorithm applied to continuous variables is used [8-9]. A population of ants can be expressed in a matrix, where the rows represent the number of design variables in the problem. Each ant is referred as an individual and has numerical values associated with it. The path of each ant is related to the value of the objective function.

The pheromone trail corresponds to an amount of pheromone laid on the path by each ant. For the i-th dimension of the design space the pheromone trail, τ_i, is given by Equation (9), where x_i^* is the i-th coordinate of the best point found by the optimization task within the design space until the current iteration, σ_i is an index related to the aggregation of the population around the current minimum for the i-th coordinate of the design space and is given by Equation (10), where **z** is a vector corresponding to the i-th column of the population matrix and $\bar{\mathbf{z}}$ is the mean value of the vector **z**.

$$\tau_i(x) = e^{-\frac{(x-x_i^*)^2}{2(\sigma_i)^2}} \tag{9}$$

$$\sigma_i = \sqrt{\frac{1}{n_{pop}-1}\sum_{j=1}^{n_{pop}}\left(z_j - \bar{z}\right)} \tag{10}$$

The updating process of the values of each design variable for all individuals is based on the probability distribution given by Equation (9). Also it can be seen that the concentration of pheromone increases in the area of the candidate to the optimum. This approach (also called as positive update) reinforces the probability of the choices that lead to good solutions. However, for avoiding premature convergence, negative update procedures are not discarded.

A simple method to perform negative update is by dissolving certain the amount of pheromone in the path. The idea of this scheme is to decrease the amount of pheromone by changing the current standard deviation (see Equation 10) for each variable. The dissolving rate affects the exploration capabilities, and consequently, the convergence of the algorithm. In the examples presented in this paper the same parameters were used in the ACO algorithm. In this investigation, the population size and the number of iterations considered for each example were equal to ten times the number of design variables.

4 DISCUSSION OF RESULTS

The methods described in the previous section were implemented using the MATLAB Code obtained from [10-11]. A finite element program [12] was used for response analysis. In this work the experimental results were synthetic which means that they were obtained from the analytical model (finite element model) considering the parameters corresponding to the damaged structure. Also noise that occurs in the experimental process was not considered. The flow between the various processors was controlled by a MATLAB main program.

This technique was applied to the steel trusses, see Figure 2 (a) and (b) for damage detection. For all the members in both structures the cross-sectional area is 12 cm² and the Young's Modulus 207 GPa. For the nine bar truss, Figure 2(a), the first four eigenmodes were used as experimental results for each damage simulation. For the fifteen bar truss, Figure 2(b) the measured response were the vertical static displacements obtained with the application of four vertical static loads of 10 kN on nodes 3, 5, 7 and 9.

The two optimizations techniques (DE and ACO) were used in the process of damage detection. Different levels of damage were simulated, from 10% until 90%, in both structures. Tables 1 and 2 show the results comparing both numeric techniques. In the nine bar truss, Figure 2(a), member 4 was damaged whereas in the fifteen bar truss, Figure 2(b), member 2 was damaged.

Table 1: Results for the nine bar steel truss.

Element	10% damage		30% damage		50% damage		70% damage		90% damage	
	DE	ACO	DE	ACO	DE	ACO	DE	ACO	DE	ACO
1	0.0005	0.0000	0.0003	0.0000	0.0000	0.0000	0.0010	0.0000	0.0028	0.0013
2	0.0000	0.0000	0.0000	0.0000	0.0005	0.0000	0.0000	0.0000	0.0087	0.0001
3	0.0009	0.0003	0.0012	0.0002	0.0003	0.0001	0.0001	0	0.0062	0.0005
4	0.1008	0.0998	0.3002	0.2997	0.4958	0.5000	0.7010	0.7000	0.9018	0.8999
5	0.0001	0.0000	0.0000	0.0000	0.0001	0.0000	0.0007	0.0000	0.0006	0.0000
6	0.0001	0.0000	0.0006	0.0002	0.0000	0.0001	0.0035	0.0002	0.0030	0.0008
7	0.0001	0.0001	0.0002	0.0002	0.0093	0.0000	0.0005	0.0001	0.0035	0.0000
8	0.0001	0.0004	0.0000	0.0003	0.0009	0.0002	0.0001	0.0001	0.0028	0.0003
9	0.0001	0.0000	0.0000	0.0000	0.0007	0.0000	0.0002	0.0000	0.0004	0.0001
Objective Function	7.7E-15	1.3E-5	2.9E-15	3.5E-6	8.8E-13	1.4-E5	4.0E-14	3.2E-6	4.5E-13	7.3E-4

Table 2: Results for the fifteen bar steel truss.

Element	10% damage		30% damage		50% damage		70% damage		90% damage	
	DE	ACO	DE	ACO	DE	ACO	DE	ACO	DE	ACO
1	0.0007	0.0000	0.0002	0.0000	0.0002	0.0000	0.0005	0.0000	0.0067	0.0000
2	0.0945	0.0939	0.2994	0.2859	0.4993	0.5055	0.6981	0.6892	0.8998	0.8999
3	0.0087	0.0026	0.0004	0.0259	0.0004	0.0000	0.0091	0.0329	0.0091	0.0285
4	0.0006	0.0000	0.0000	0.0213	0.0001	0.0000	0.0137	0.0200	0.0007	0.0040
5	0.0005	0.0000	0.0001	0.0000	0.0008	0.0000	0.0000	0.0000	0.0016	0.0010
6	0.0002	0.0326	0.0006	0.0416	0.0002	0.0422	0.0006	0.0418	0.0184	0.0154
7	0.0037	0.0013	0.0000	0.0199	0.0002	0.0152	0.0003	0.0263	0.0000	0.0086
8	0.0149	0.0000	0.0019	0.0000	0.0262	0.0000	0.0470	0.0003	0.0029	0.0019
9	0.0001	0.0000	0.0001	0.0000	0.0002	0.0000	0.0044	0.0000	0.0002	0.0010
10	0.0000	0.0001	0.0000	0.0000	0.0005	0.0000	0.0013	0.0000	0.0042	0.0004
11	0.0103	0.0002	0.0001	0.1313	0.0000	0.0055	0.0163	0.2449	0.0185	0.0109
12	0.0004	0.0000	0.0000	0.0000	0.0000	0.0000	0.0027	0.0000	0.0003	0.0001
13	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0014	0.0000	0.0000	0.0001
14	0.0007	0.0000	0.0000	0.0000	0.0010	0.0000	0.0000	0.0001	0.0000	0.0002
15	0.0001	0.0000	0.0000	0.0233	0.0001	0.0006	0.0067	0.1018	0.0095	0.0000
Objective Function	2.2E-11	1.3E-8	8.3E-13	8.3E-9	6.1E-12	1.2E-8	3.1E-10	7.7E-9	4.1E-10	1.2E-8

It must be emphasized that in general, both methods performed very well. The location and level of damaged in all cases were detected correctly. The Differential Evolution was more efficient. The results were more accurate with less computational effort. In some cases the Ant Colony Optimization detected damage in good members as occurred in member 11 of the fifteen bar truss for some cases of damage.

5 CONCLUSIONS

The paper presented an approach for damage detection using a system identification technique that results in a nonlinear optimization problem. Two global optimization methods were used: Differential Evolution (DE) and Ant Colony Optimization (ACO). Reduced set of eigenmodes and static displacements were used as experimental data in the identification procedure. The methods were applied to simple steel truss structures with different levels of damage and presented promising results. Both optimization techniques performed well identifying the location and extent of damage very clearly. The Differential Evolution method was more efficient presenting more accurate results with less computational effort. An extension of this work is to study the application of the present approach to large truss structures and to other types of structures such as beams, plates and shells.

6 ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support provided by the Brazilian National and State Scientific and Technological Supporting Agencies: CNPq, CAPES and FAPERJ.

REFERENCES

[1] Hajela, P. and Soeiro, F. J. C. P., "Recent Developments in Damage Detection Based on System Identification Methods", Structural Optimization, 1990.

[2] Hajela, P. and Soeiro, F. J. C. P., "Structural Damage Detection Based on Static and Modal Analysis", AIAA Journal, 1990.

[3] Soeiro, F. J. C. P. and Hajela, P., "Locating Damage in a Truss Structure Using Identification Techniques: an Experimental Investigation", Journal of the Brazilian Society of Mechanical Sciences, 1992.

[4] Storn, R. and Price, K., "Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", Journal of Global Optimization, Kluwer Academic Publishers, 1997.

[5] Storn, R., Price, K. and Lampinen, J., "Differential Evolution - A Practical Approach to Global Optimization", Springer-Verlag, Natural Computing Series, Berlin Heidelberg, 2005.

[6] Dorigo, M., "Optimization, Learning and Natural Algorithms", PhD thesis, Politecnico di Milano, Italy, 1992,

[7] Dorigo, M. and Gambardella, L. M., "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem", IEEE Transactions on Evolutionary Computation, 1997.

[8] Pourtakdoust, S. H. and Nobahari, H., "An Extension of Ant Colony System to Continuous Optimization Problems", Lecture Notes in Computer Science, 2004.

[9] Socha, K., "ACO for Continuous and Mixed-Variable Optimization", Lecture Notes in Computer Science, 2004.

[10] Storn, R. and Price, K., "Differential Evolution (DE) for Continuous Function Optimization", http://www.icsi.berkeley.edu/~storn/code.html, 2009.

[11] Viana, F.A.C., "Simple Optimization Toolbox", <http://fchegury.googlepages.com/>, University of Florida, 2008.

[12] Kwon, Y. W. and Bang, H. C., "The Finite Element Method Using MATLAB", 2nd Ed. CRC Press, Boca Raton, Fl., 2000.