UNIVERSIDAD NACIONAL DE COLOMBIA
SEDE MEDELLÍN
FACULTAD DE MINAS

DYNA
http://dyna.medellin.unal.edu.co/

# Assessing the performance of a differential evolution algorithm in structural damage detection by varying the objective function

Jesús Daniel Villalba-Morales[a] & José Elias Laier [b]

[a] *Facultad de Ingeniería, Pontificia Universidad Javeriana, Bogotá, Colombia. jesus.villalba@javeriana.edu.co*
[b] *Escuela de Ingeniería de São Carlos, Universidad de São Paulo, São Paulo Brasil. jelaier@sc.usp.br*

Received: December 9th, 2013. Received in revised form: March 11th, 2014. Accepted: November 10 th, 2014.

**Abstract**
Structural damage detection has become an important research topic in certain segments of the engineering community. These methodologies occasionally formulate an optimization problem by defining an objective function based on dynamic parameters, with metaheuristics used to find the solution. In this study, damage localization and quantification is performed by an Adaptive Differential Evolution algorithm, which solves the associated optimization problem. Furthermore, this paper looks at the proposed methodology's performance when using different functions based on natural frequencies, mode shapes, modal flexibilities, modal strain energies and the residual force vector. Simple and multiple damage scenarios are numerically imposed on truss structures to assess the performance of the proposed methodology. Results show that damage scenarios can be reliably determined by using the analyzed objective functions. However, the methodology does not perform well when the objective function based on natural frequencies and modal strain energies is employed.

*Keywords*: damage detection; metaheuristics; optimization; dynamic parameters; differential evolution.

# Valoración del desempeño de un algoritmo de evolución diferencial en detección de daño estructural considerando diversas funciones objetivo

**Resumen**
Detección de daño estructural es actualmente un importante tema de investigación para diferentes comunidades en ingeniería. Algunas de las metodologías de detección de daño reportadas en la literatura formulan un problema de optimización mediante una función objetivo basada en la respuesta dinámica de la estructura y el uso de metaheurísticas para resolverlo. En este estudio, la localización y cuantificación del daño se realiza utilizando un algoritmo de evolución diferencial con parámetros adaptativos. El desempeño de la metodología propuesta es evaluado utilizando diversas funciones objetivo basadas en frecuencias naturales, formas modales, flexibilidad modal, energía de deformación modal y el vector de fuerza residual. Escenarios de daño simple y múltiple son simulados para estructuras de tipo armadura con el objetivo de determinar el desempeño de la metodología propuesta. Los resultados muestran que el algoritmo utilizado puede determinar confiablemente los escenarios buscados para varias de las funciones objetivo utilizadas. Sin embargo, no se obtuvo buenos resultados cuando se utilizó la función basada en frecuencias naturales y energía de deformación modal.

*Palabras clave*: detección de daño; metaheurísticas; optimización parámetros dinámicos; evolución diferencial.

## 1. Introduction

Metaheuristics are a set of computational techniques that help find suitable solutions for optimization problems reasonably quickly. These computational techniques are characterized by the ease of their design and implementation [1]; they are primarily used to tackle complex optimization problems that do not count on a specific algorithm for their solution, such as the damage detection problem. When it comes to finding a solution for this type of problem, metaheuristics often prove to be the most effective tool [2]. Using metaheuristics to solve this problem offers several advantages over other, more conventional optimization technique—the ability to find a global solution, lower

dependence on the initial solution, less sensitivity to noise in measurements and freedom from computing derivatives to guide the search [3]. On the flip side, metaheuristics´ drawbacks include the definition of their own parameters, e.g. crossover and mutation rates for genetic algorithms, a process generally carried out by trial and error.

Previous studies have demonstrated the potential of using metaheuristics for structural damage detection [3-8]. However, no study has defined the best metaheuristic to be used. Consequently, determining the performance of metaheuristics to detect damage proves to be a very interesting topic of research. One of the most promising metaheuristics is the Differential Evolution Algorithm (DE), which solves real-value optimization problems by evolving a population of possible solutions. Only a few damage detection methodologies, such as [7], rely on the DE approach. That proposal required analysis in sub-domains to improve the methodology performance. As damage representation implies one optimization variable for each structural element, the division into sub-domains reduced the number of optimization variables for this problem. In addition to damage identification, DE has been put to use in the field of structural engineering, such as structural identification [9] and structural optimization [10]. The successful application to the aforementioned type of problem turns DE into an appealing alternative for damage identification.

Using DE entails the definition of a few parameters: population size (NP), crossover rate (CR) and amplification factor (AF), all three of which control the evolutionary process. While papers that propose methodologies to control DE parameters abound in the literature, such as reference [11,12], there is not sufficient research that focuses on metaheuristics that do not depend on their parameters for damage detection.

This paper is aimed at addressing this lack by evaluating the performance of an adaptive differential algorithm to detect structural damage. A simple method of adaptation is used to avoid defining DE parameters by trials, leaving NP as the only uncontrolled parameter. A second main contribution of this paper relates to the definition of the objective function. That is to say, the possibility of selecting different dynamic parameters to form the objective function permits the determination of the best parameters for use. To that end, several objective functions are proposed, which are based on dynamic parameters such as natural frequencies, modal flexibilities, modal strain energies and residual force vectors. Methodology's performance is assessed in two parts. First, it is applied to the detection of simulated simple and multiple damage scenarios in truss structures. Then, the results for each function are compared in terms of the differences between computed and simulated damage scenarios to choose the parameters best suited for the damage detection problem in trusses.

## 2 Adaptive Differential Evolutionary Algorithm

De [13] is an algorithm used to solve optimization problems involving continuous domains. The process of DE for optimization problems is summarized in Fig. 1.

**Begin**

1. Generate initial vectors.
2. Compute cost of each vector.
3. Run from i=1 to NP
   - Define the i-th vector in the population as Target Vector.
   - Create Trial Vector.
   - Compute cost of Trial Vector.
   - Compare Trial and Target Vectors; choose the best one.
4. Verify convergence criteria. If unmet, repeat from Step 2. Otherwise, go to Step 5.
5. Show best solution vector in population.

**End**

Figure 1. Differential Evolution Algorithm.
Source: The authors

The algorithm iteratively modifies a set of solutions (population) by means of specific operators to find the best solution. Each solution in the population is called a solution vector. A value for NP must be defined in Step 1, with initial solutions either heuristically or randomly generated. The quality (read: cost) of each initial solution (Step 2) is quantified by using the objective function. For a maximization problem, the higher an objective function's value, the higher the quality of the solution. Step 3 compares the cost of a Target Vector (selected from the current population) and a new vector, referred to as a Trial Vector. The vector with the highest cost will form part of the population in the next iteration; this process is carried out for each vector in the current population, one vector at a time. After all vectors in the population have been exhausted, DE moves on to Step 4, which consists of verifying the convergence criteria. If these criteria are not reached, Step 3 is carried out again. Otherwise, the algorithm advances to the next step. Finally, the solution to the problem is revealed.

The creation of trial vectors is laid out below:

First, a mutation vector is created from the variation of the current best vector [14]:

$$\boldsymbol{v}_i^t = \boldsymbol{x}_{best}^t + AF \cdot V_{amp} \tag{1}$$

where I is the position of the target vector in the population, xbest is the best solution vector in the current iteration t. AF is a DE parameter that allows for control of the variation amplitude introduced by Vamp, with Vamp being a quantity vector computed as [15]

$$V_{amp} = \sum_{k=1}^{4} ((-1)^{k+1} \cdot \boldsymbol{x}_k^t) \tag{2}$$

where $x_k$ are vectors in the current population, which are randomly chosen. Not only must these $x_k$ vectors be different from each other, but they must also be different from the current best solution vector.

Next, the *i*-th trial vector can be computed using the binary crossover [14]:

$$\boldsymbol{u}_i^t = \{\boldsymbol{u}_{i,1} \quad \cdots \quad \boldsymbol{u}_{i,n}\}^t \tag{3}$$

Each term in the trial vector is assigned as

$$\boldsymbol{u}_{i,j}^t = \begin{cases} \boldsymbol{v}_{i,j}^t & if\ nr_j < CR \\ \boldsymbol{tg}_{i,j}^t & if\ nr_j \geq CR \end{cases} \tag{4}$$

where *tg* is the target vector, $nr_j$ is a random number generated for each position *j* of the *i*-th vector, which ranges between 0 and 1. The latter parameter controls the role played by both mutation and target vectors in defining the trial vector.

Values for *cr* and *af* must be previously assigned before employing the DE. An iterative process that sets DE parameters is frequently relied on to define the most appropriate values. To avoid this process, the present study gives each solution vector its own *cr* and *af* parameters. These values can evolve throughout the DE´s execution, and their initial values are randomly generated between for *af* [0.7-0.9] and *cr* [0.4-0.6]. Far from arbitrary, these ranges are chosen because they account for the recommended values of 0.8 and 0.5 for *af* [15] and *cr* [16], respectively. *cr* and *af* values for the iteration *t*+1 can be computed as

$$\boldsymbol{cr}_i^{t+1} = \boldsymbol{cr}_{best}^t + \boldsymbol{R}_i\ (\boldsymbol{cr}_{best}^t - \boldsymbol{cr}) \tag{5}$$

$$\boldsymbol{af}_i^{t+1} = \boldsymbol{af}_{best}^t + \boldsymbol{R}_i\ (\boldsymbol{af}_{best}^t - \boldsymbol{af}_i^t) \tag{6}$$

where *best* corresponds to the best vector in the current population. $R_{i,1}$ and $R_{i,2}$ are random numbers between 0 and 1 generated for solution vector *i*.

Concerning the definition of *NP*, Storn and Price [15] proposed a size between 5 and 10 the number of design variables. However, NP definition utilized in this research will be explained in Section 4.

## 3. Proposed damage detection methodology

The methodology employed throughout the present paper (Fig. 2) can be consulted in [17], with one notable exception: for our purposes, the genetic algorithm has been replaced by a DE.

This methodology assumes the existence of a finite element model (FEM) that is able to accurately represent the undamaged condition (Step 1). Damage takes the form of a reduction in the stiffness matrix of the damaged element by means of a stiffness reduction factor β. A value equal to 0 for this factor indicates no damage and 1 total damage.

**Begin**
1. Define the finite-element model of initial structure.
2. Determine dynamic parameters of current structure.
3. Define the objective function.
4. Apply DE algorithm
5. Show damage scenario found.

**End**

Figure 2. Damage Detection Methodology.
Source: Adapted from Villalba, J. D. and Laier, J. E.2012

Optimization aims to arrive at the correct set of β factors, i. e., those which allow the dynamic parameters of an updated FEM to match those measured experimentally for the current structure. Each possible solution vector corresponds to a different damage scenario, and each position in the vector represents the β factor of one structural element. For an undamped structure with a mass matrix *M* and a stiffness matrix *K*, the modal parameters are given by

$$\left(\boldsymbol{K} - \omega_j^2 \cdot \boldsymbol{M}\right)\{\phi\}_j = 0 \tag{7}$$

where $\omega_j$ is the natural frequency corresponding to the *j*-th mode shape $\phi_j$. Thus, any change in the stiffness or mass matrix will lead to changes in the dynamic parameters.

Step 2 experimentally determines the structure's dynamic behavior. As this is a numerical study, the current dynamic parameters are obtained by introducing the simulated damage scenario into the undamaged structure's FEM. Then, the eigen-value problem, given in Eq. 7, is solved. These parameters are numerically perturbed to simulate noise in a real measurement. Such perturbations equal 1% for the natural frequencies ($noise_\omega$) and 3% for the mode shapes ($noise_\Phi$) [18]. The equations that introduce noise in the modal parameters are [3]:

$$\begin{aligned} \Phi_{ij}^n &= \Phi_{ij} \cdot \left(1 + rand \cdot noise_\phi\right) \\ \omega_j^n &= \omega_j \cdot (1 + rand \cdot noise_\omega) \end{aligned} \tag{8}$$

where *Rand* is a random number between -1 and 1. Letter *n* means a parameter with noise.

Armed with the available modal data, an objective function must be defined (Step 3), which generally has the following form:

$$Min\ F = curr\_dyn\_resp - upd\_dyn\_resp \tag{9}$$

where *curr_dyn_resp* and *upd_dyn_resp* refer to the dynamic parameters for the current structure and updated model, respectively. A detailed version of the objective functions relied on to carry out this research appears in Section 3.1.

Step 4 applies the DE described in Section 2 in tandem with the heuristic shown in Section 3.2. In the same vein, convergence criteria are presented along with the numerical example because the maximum number of iterations depends on the structure size.

Finally, the computed damage scenario is reported in the form of stiffness reduction factor ($\beta$) value for each of the 61 elements in the structure.

### *3.1. Objective Functions*

In this study, the damage detection problem was treated as a maximization one that is equivalent to the minimization formulation in Eq 9. The objective functions below were those used for result comparison.

- Natural Frequencies and Mode Shapes (10):

$$F_1 = \sum_{j=1}^{nm} \frac{c_1}{c_2 + \left|\frac{\boldsymbol{\omega}_j^{dea} - \boldsymbol{\omega}_j^{ex}}{\boldsymbol{\omega}_j^{ex}}\right| + w_1 \cdot \sqrt{\frac{\sum_{i=1}^{ngll}\left(\boldsymbol{\Phi}_{ij}^{dea} - \boldsymbol{\Phi}_{ij}^{ex}\right)^2}{\sum_{i=1}^{ngll}\left(\boldsymbol{\Phi}_{ij}^{ex}\right)^2}}}$$

- Natural Frequencies and Strain Modal Energies (11)

$$F_2 = \sum_{j=1}^{nm} \frac{c_1}{c_2 + \left|\frac{\boldsymbol{\omega}_j^{dea} - \boldsymbol{\omega}_j^{ex}}{\boldsymbol{\omega}_j^{ex}}\right| + w_1 \cdot \left|\frac{\boldsymbol{mse}_j^{dea} - \boldsymbol{mse}_j^{ex}}{\boldsymbol{mse}_j^{dea}}\right|}$$

- Residual Force Vector (12)

$$F_3 = \sum_{j=1}^{nm} \frac{c_1}{c_2 + \frac{\boldsymbol{RFV}_j^{dea} \times \boldsymbol{RFV}_j^{dea}}{\boldsymbol{RFV}_j^{ex} \times \boldsymbol{RFV}_j^{ex}}}$$

- Modal Flexibility (13)

$$F_3 = \sum_{j=1}^{nm} \frac{c_1}{c_2 + \frac{\sum_{j=1}^{ngll}\sum_{i=1}^{ngll}\left|\boldsymbol{MF}_{ij}^{dea} - \boldsymbol{MF}_{ij}^{ex}\right|}{\sum_{j=1}^{ngll}\sum_{i=1}^{ngll}\left|\boldsymbol{MF}_{ij}^{ex}\right|}}$$

The symbols used in the above functions are: *nm* is the number of modes available, $c_{1,2}$ are constants defined as 200 and 1, respectively, $\omega_j$ is the *j*-th natural frequency, $\phi_{ij}$ is a value for the *i*-th degree of freedom of the *j*-th mode shape, *dea* is a value from the DE solution and *ex* is experimental information, *ngll* is the number of degrees of freedom of the structure, $W_{1,2}$=2.0 weight factor, *und* refers to the undamaged structure, $mse_j$ is the modal strain energy for the *j*-th mode shape and $MF_{ij}$ is a value corresponding to the position (*i*,*j*) of the modal flexibility.

The terms $MSE_j$, *RFV* and *MF* are given by

$$\boldsymbol{mse}_j = \boldsymbol{\Phi}_j^T \times \boldsymbol{K} \times \boldsymbol{\Phi}_j \tag{14}$$

$$\boldsymbol{RFV} = \left(\boldsymbol{K} - \omega_j^2 \cdot \boldsymbol{M}\right) \times \boldsymbol{\Phi} \tag{15}$$

$$\boldsymbol{MF} = \sum_{j=1}^{nm} \frac{1}{\omega_j^2} \boldsymbol{\Phi}_j \boldsymbol{\Phi}_j^T \tag{16}$$

All functions are based on the difference between the dynamic parameters computed for the FEM of a specific vector solution –possible damage scenario– and those obtained experimentally. Thus, each possible damage scenario in the population displays a given probability of being the correct one. The objective function is put into use one at a time, so the performance of the proposed methodology using this function can be measured.

### *3.2. Heuristic for the generation of the initial population vector*

To accelerate DE convergence, the initial population was generated heuristically based on two characteristics assumed for the damage scenarios. First, damaged element quantity is considered low in light of its relation to the total number of elements in the structure [19]. Second, damage extent is not expected to be severe. The proposed heuristic generates a random number between 0 and 1 for each $\beta_i$ factor in each initial solution vector. If the random number is smaller than 0.5, then $\beta_i$ takes a random value between 0.1 and 0.5; otherwise, $\beta_i$ is set to zero.

## 4. Numerical Examples

The proposed methodology was applied to detect damage in truss structures with different configurations (Fig. 3). The FEM uses conventional 2D bar elements assuming perfectly pinned joints. Vertical and horizontal elements have a length equal to 1.0m. Elements have elastic modulus E=200 x $10E^9$ N/m$^2$, density ρ=7800 kg/m$^3$ and cross-sectional area A=0.001 m$^2$. Information regarding the first eight natural frequencies and complete mode shapes of the damaged structure were considered available.

Applying the proposed methodology to a total of 21 simulated damage scenarios (see Tables 1-4, where DS is a damage scenario identifier) makes it possible to evaluate the methodology's performance. Initially, results for the 61-element truss will be used to compare how well the four proposed objective functions work.

Tables 1 and 2 include identified damaged elements and damage extents (βi) for simulated damage scenarios. Scenarios seen in Table 1 correspond to damage present in only one element, whether the damage be in a vertical (DS-1, DS-4 and DS-7), horizontal (DS-2, DS-5, DS-8) or diagonal (DS-3, DS-6, and DS-9) element. Multiple damaged elements are considered in scenarios DS-10, DS-11 and DS-12 (Table 2). The focus of the results will be on the correct identification of damaged elements and the presence of misidentified elements. The objective functions producing the best damage identification are then used to detect damage in the other trusses. Simulated damage scenarios for these trusses are shown in Table 3-5, which include damage located in a specific zone of the truss or spread out along the structure. The reason for using only cases of multiple damage scenarios is brought up later. In this order of ideas, more detail-oriented analysis for these results will be undertaken in section 5.

Table 1.
Simulated simple damage scenarios for the 61-element truss structure.

| Damage Scenario | Damaged Element | Damage Extent (β) |
|---|---|---|
| DS-1 | 3 | 0.15 |
| DS-2 | 24 | 0.20 |
| DS-3 | 52 | 0.40 |
| DS-4 | 6 | 0.20 |
| DS-5 | 33 | 0.20 |
| DS-6 | 55 | 0.20 |
| DS-7 | 10 | 0.20 |
| DS-8 | 27 | 0.20 |
| DS-9 | 50 | 0.20 |

Source: The authors.

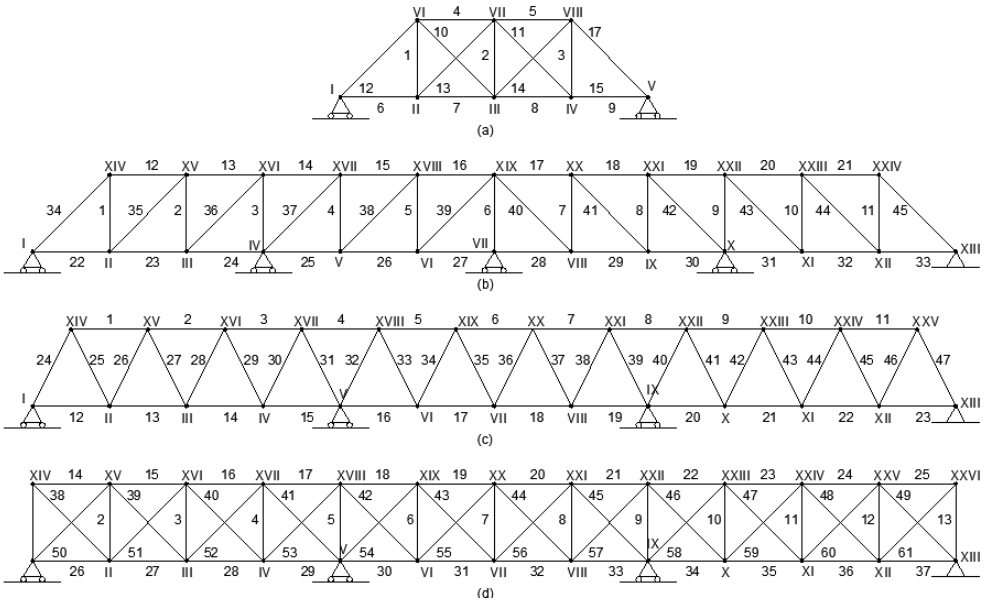

Figure 3. Analyzed structures. a) 15, b) 45, c) 47 and d) 61 element trusses.
Source: Adapted from Villalba, J. D. and Laier, J. E.2012.

Table 2.
Simulated multiple damage scenarios for the 61-element truss structure.

| DS-10 | | DS-11 | | DS-12 | |
|---|---|---|---|---|---|
| Elem | βi | Elem | βi | Elem | βi |
| 3 | 0.150 | 4 | 0.250 | 3 | 0.170 |
| 28 | 0.150 | 15 | 0.200 | 10 | 0.210 |
| 52 | 0.150 | 18 | 0.230 | 14 | 0.140 |
| | | 24 | 0.220 | 17 | 0.180 |
| | | 34 | 0.260 | 37 | 0.200 |
| | | | | 52 | 0.200 |

Source: The authors.

Table 3.
Simulated multiple damage scenarios for the 15-element truss structure.

| DS-13 | | DS-14 | | DS-15 | |
|---|---|---|---|---|---|
| Elem | βi | Elem | βi | Elem | βi |
| 3 | 0.26 | 4 | 0.24 | 2 | 0.25 |
| 8 | 0.22 | 10 | 0.28 | 13 | 0.25 |
| | | 13 | 0.31 | 15 | 0.25 |

Source: The authors.

Table 4.
Simulated multiple damage scenarios for the 45-element truss structure.

| T16 | | T17 | | T18 | |
|---|---|---|---|---|---|
| Elem | βi | Elem | βi | Elem | βi |
| 5 | 0.32 | 1 | 0.23 | 7 | 0.30 |
| 26 | 0.29 | 13 | 0.31 | 23 | 0.30 |
| 38 | 0.24 | 32 | 0.26 | 35 | 0.30 |
| | | 41 | 0.18 | 41 | 0.30 |

Source: The authors.

In addition to the structure's physical characteristics, terms related to the DE execution require definition. All DE parameters were defined for the 61-element structure before being brought to bear on the other structures. First, observation shows that NP depends on the number of optimization variables. Yet, the relationship between NP and structure size has not quite been elucidated for the damage detection problem. As a result, NP was set at 200 after a process of trial and error. The criterion for halting the algorithm's execution is either reaching a maximum of 200 iterations or 50 iterations without significant changes in the best current vector's cost. 30 executions were done for each example studied to ensure the proposed methodology reliably detects the real damage scenario. The computed damage scenario that corresponds to the run with the highest cost solution is then reported to the user.

Table 5.
Simulated multiple damage scenarios for the 47-element truss structure.

| T19 | | T20 | | T21 | |
|---|---|---|---|---|---|
| Elem | βi | Elem | βi | Elem | βi |
| 3 | 0.26 | 4 | 0.24 | 2 | 0.25 |
| 8 | 0.22 | 10 | 0.28 | 10 | 0.25 |
| | | 13 | 0.31 | 14 | 0.25 |

Source: The authors.

## 5. Results and Discussion

Results for cases with only one damaged element in the 61-element structure are shown in Figs. 4-6. The methodology detects the real damaged element and damage extent accurately. Functions F1 and F4 provided the best results across all cases analyzed. They properly gauged damage extent, with the difference between real and simulated damage less than 0.08. Moreover, few elements were misidentified. The function based on the residual force vector (F3) misidentified elements on numerous occasions. Naturally, these errors translate into low methodology reliability when function F3 is used. For all cases, reliance upon an objective function based on natural frequencies and modal strain energy left the real damaged element undiscovered.

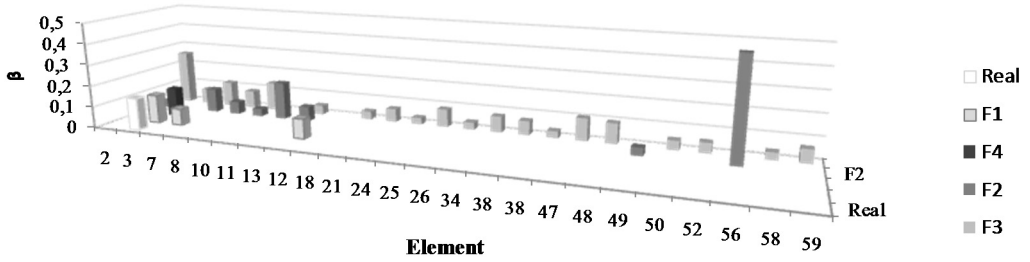


Figure 4. Results for Damage Scenario DS-1.
Source: The authors.

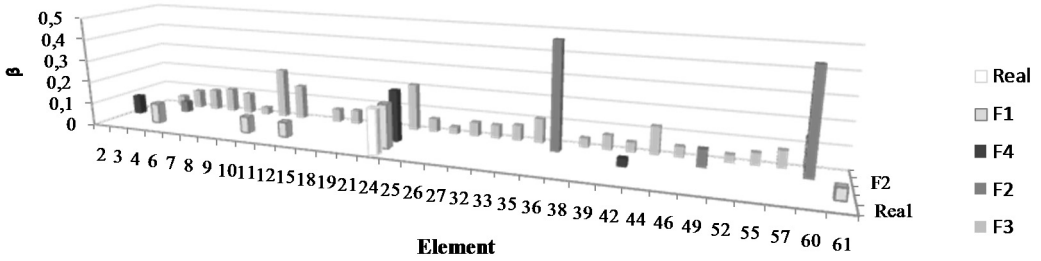


Figure 5. Results for Damage Scenario DS-2.
Source: The authors.

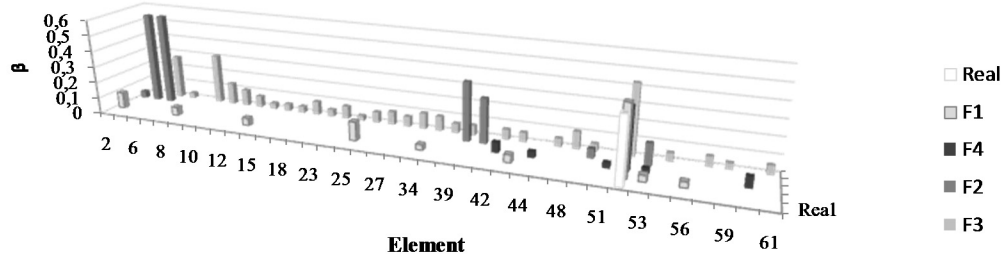


Figure 6. Results for Damage Scenario DS-3.
Source: The authors.

Table 6 presents methodology performance when an energy-based objective function was used to detect simple damage in the 61-element truss. Results evince the low reliability when using Function F2, for there is a significant gap between real and computed damage extents, as well as many misidentified damage elements. Table 7 confirms this low reliability by displaying the best five runs for a scenario wherein Element 50 is damaged (only $\beta_i$ factors $\geq 0.05$ are shown): not a single computed scenario included the real damaged element. The algorithm could not distinguish between completely different solutions - scenarios with varying damaged elements and damage extents - and converge on any of these solutions. Therefore, the results for the detection of multiple damage scenarios using function F2 will not be shown.

Table 6.
Performance of the proposed methodology to detect simple damage using an objective function based on strain modal energy.

| DS-4 | | | DS-5 | | | DS-6 | | | DS-7 | | | DS-8 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Element | $\beta_{real}$ | $\beta_{comp}$ | Element | $\beta_{real}$ | $\beta_{comp}$ | Element | $\beta_{real}$ | $\beta_{comp}$ | Element | $\beta_{real}$ | $\beta_{comp}$ | Element | $\beta_{real}$ | $\beta_{comp}$ |
| 2 | --- | 0.54 | 3 | --- | 0.10 | 2 | --- | 0.06 | 10 | 0.20 | 0.05 | 2 | --- | 0.07 |
| 6 | 0.20 | 0.04 | 8 | --- | 0.05 | 11 | --- | 0.53 | 12 | --- | 0.60 | 4 | --- | 0.05 |
| 12 | --- | 0.51 | 11 | --- | 0.53 | 12 | --- | 0.59 | 34 | -- | 0.09 | 7 | --- | 0.15 |
| 36 | --- | 0.41 | 33 | 0.20 | 0.00 | 31 | --- | 0.1 | | | | 12 | --- | 0.57 |
| | | | 36 | --- | 0.49 | 39 | --- | 0.07 | | | | 14 | --- | 0.09 |
| | | | 44 | --- | 0.35 | 55 | 0.20 | --- | | | | 27 | 0.20 | --- |
| | | | 56 | --- | 0.10 | | | | | | | 36 | --- | 0.37 |
| | | | | | | | | | | | | 44 | --- | 0.15 |
| | | | | | | | | | | | | 52 | --- | 0.09 |
| | | | | | | | | | | | | 60 | --- | 0.09 |

Source: The authors.

Table 7.
Comparison of different solutions found for damage scenario DS-9 with the objective function based on strain modal energy.

| Element | 1 | 2 | 7 | 8 | 11 | 13 | 25 | 34 | 35 | 36 | 47 | 50 | 59 | 60 | Cost |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\beta_{real}$ | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | 0.20 | --- | --- | --- |
| $\beta_{sol1}$ | --- | --- | --- | 0.05 | --- | 0.06 | --- | --- | --- | 0.40 | --- | --- | --- | 0.39 | 1350.4 |
| $\beta_{sol2}$ | --- | --- | --- | --- | --- | --- | --- | --- | 0.37 | --- | 0.35 | --- | --- | --- | 1343.8 |
| $\beta_{sol3}$ | --- | 0.05 | 0.06 | 0.08 | 0.48 | --- | 0.39 | 0.03 | --- | --- | --- | --- | 0.05 | 0.12 | 1329.3 |
| $\beta_{sol4}$ | --- | --- | --- | --- | 0.50 | --- | --- | 0.06 | --- | --- | --- | --- | 0.39 | --- | 1325.8 |
| $\beta_{sol5}$ | 0.09 | 0.19 | 0.22 | 0.17 | 0.35 | --- | 0.41 | 0.03 | --- | --- | --- | --- | --- | 0.07 | 1319.3 |

Source: The authors.

Results obtained for simulated multiple damage scenarios in the 61-element structure are shown in Figs. 7-9. The algorithm detected all real damaged elements when objective functions F1, F3 and F4 were used. With respect to the damage extent, the difference between computed and real extent is less than 0.1 for every element in all three scenarios. As was the case for simple damage scenarios, objective function F3 produced the most misidentified elements. Though, as previously mentioned, most functions included misidentification, Function F1 identified only the real damaged elements for Scenario DS-10 (Fig. 7). Function F3 misidentified 40% of the elements for scenario DS-11, and seven of these misidentified elements exhibited a damage extent higher than 0.1. Therefore, the logical conclusion is that objective function F3 is less reliable than the others. For scenario DS-12, Function F4 produced the best results, with an average difference of 0.047 between simulated and real damage extent for all real damage elements.

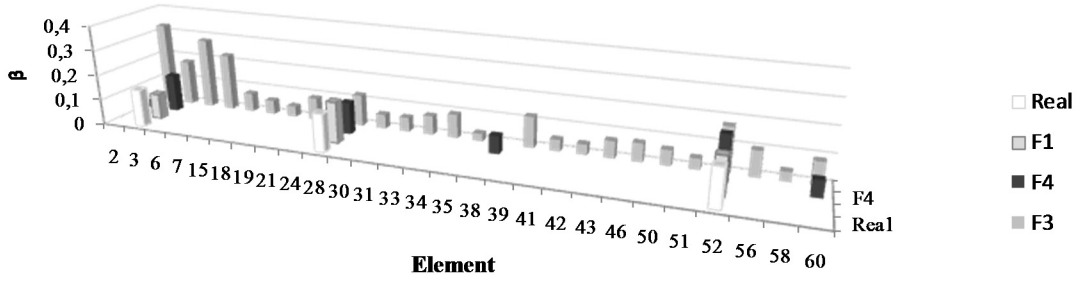


Figure7. Results for Damage Scenario DS-10.
Source: The authors.

In order to more thoroughly understand the methodology's effectiveness in terms of computing real damage extent, readers are guided to Tables 8-10, which lay out the differences between real ($\beta_{real}$) and computed ($\beta_{comp}$) damage extent. The value denoted in parentheses indicates such difference in terms of percentage. Looking at these results, it can be inferred that as the number of damaged elements increases, error extent also increases. For example, when using Function F4, the average error was 11, 24 and 25 % for 3, 5 and 6 damaged elements, respectively. In spite of this fact, results for Functions F1 and F4 do not display a significant difference between the computed and real damage extent. Function F2 presents an average error of 19% for the three damage scenarios. Then there is the case of Function F4, which underestimated the damage extent for element 14 in scenario DS-12. It is important to recall that element 14 had a low real damage value. In this sense, dynamic parameter changes caused by low damage values can be masked by noise in the measurements.

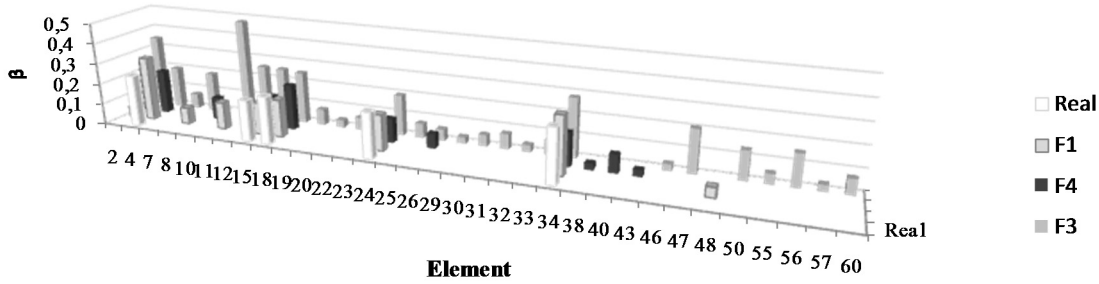


Figure 8. Results for Damage Scenario DS-11.
Source: The authors.

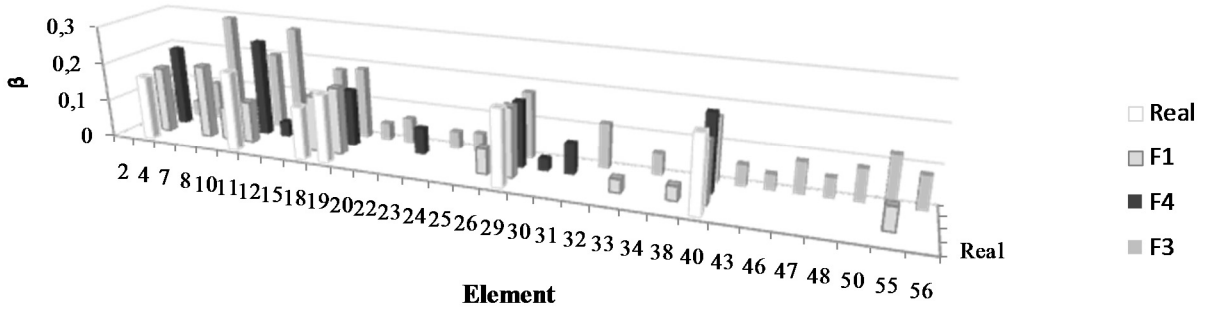


Figure 9. Results for Damage Scenario DS-12.
Source: The authors.

Table 8.
Comparing the real and computed damage extent for scenario DS-10 using functions F1, F3, F4.

| Element | $\beta_{real}$ | $\beta_{comp}$ | | |
|---|---|---|---|---|
| | | $F_1$ | $F_3$ | $F_4$ |
| 3 | 0.15 | 0.10 (33) | 0.18 (20) | 0.15 (0) |
| 28 | 0.15 | 0.16 (7) | 0.13(13) | 0.13 (13) |
| 52 | 0.15 | 0.15(0) | 0.18(20) | 0.18 (20) |

Source: The authors.

Table 9.
Comparing the real and computed damage extent for scenario DS-11 using functions F1, F3, F4.

| Element | $\beta_{real}$ | $\beta_{comp}$ | | |
|---|---|---|---|---|
| | | $F_1$ | $F_3$ | $F_4$ |
| 4 | 0.25 | 0.31 (24) | 0.20 (20) | 0.21 (16) |
| 15 | 0.2 | 0.16 (20) | 0.26 (30) | 0.15(25) |
| 18 | 0.23 | 0.18 (22) | 0.25 (9) | 0.22 (4) |
| 24 | 0.22 | 0.17 (23) | 0.20 (9) | 0.12 (45) |
| 34 | 0.23 | 0.28 (22) | 0.28 (22) | 0.16 (30) |

Source: The authors.

Table 10.
Comparing the real and computed damage extent for scenario DS-12 using functions F1, F3, F4.

| Element | $\beta_{real}$ | $\beta_{comp}$ | | |
|---|---|---|---|---|
| | | $F_1$ | $F_3$ | $F_4$ |
| 3 | 0.17 | 0.18 (6) | 0.03 (82) | 0.21(24) |
| 10 | 0.21 | 0.11(48) | 0.20 (5) | 0.26 (24) |
| 14 | 0.14 | 0.15 (7) | 0.18 (29) | 0.04 (71) |
| 17 | 0.18 | 0.18 (0) | 0.19 (6) | 0.15 (17) |
| 37 | 0.2 | 0.18 (10) | 0.18 (10) | 0.17 (15) |
| 52 | 0.2 | 0.16 (20) | 0.17 (15) | 0.20 (0) |

Source: The authors.

Table 11 shows the number of executions for which the algorithm found all real damaged elements. Here, one should keep in mind that the proposed DE performs 30 runs to detect each damage scenario. Based on the results of this study, each objective function possesses a different reliability level; however, it is safe to say objective function F4 (modal flexibility) was the group's weak link. Despite the high success rate of detecting real damaged elements with F3, results for this function also presented many misidentified elements. On the whole, simple damage scenarios were detected more reliably than multiple damage scenarios, regardless of the objective function employed. In response to this fact, the performance assessment for the other trusses entailed multiple damage scenarios.

Table 11.
DE Performance when identifying all real damaged elements.

| Damage Scenario | Number of successful executions | | |
|---|---|---|---|
| | F1 | F3 | F4 |
| DS-1 | 30 | 23 | 26 |
| DS-2 | 30 | 29 | 25 |
| DS-3 | 30 | 30 | 18 |
| DS-4 | 15 | 15 | 8 |
| DS-5 | 19 | 18 | 8 |
| DS-6 | 7 | 19 | 8 |

Source: The authors.

As the function F3 reported many misidentified elements, results for the other trusses will only include Functions F1 and F4. Tables 12-14, 15-17 and 18-20 depict the application of the proposed methodology for 15, 45, 47-element trusses, respectively. Reported damage scenarios only include those elements presenting β 0.05. As previously mentioned, the percentage difference between the computed and real damage extent is placed in parentheses: results show that the methodology precisely computes damage extent for both functions. The exact scenario DS-13 was obtained with Function F1 for the smallest truss structure. It is worth mentioning that the search space correlates to the number of elements in the truss structure; thus, as the number of elements increases, the search space increases, adding another level of complexity to the optimization process. Detecting element 2 in Scenario DS-15 was difficult for function F4, exposing the difficulty to detect damage in some elements. While the average error for all the scenarios is 4.2 and 14.3 for function F1 and F4, respectively. Function F4 means more misidentified elements compared with the results obtained using F1. These elements presented low damage extent for both functions in most cases. However, the existence of misidentified elements with high damage values is possible, i. e., Element 5 in Scenario DS-17 as computed with Function F1 or Element 10 in Scenario DS-15 as computed with Function F4. Overall, F1 and F4 produce similar results, albeit with greater reliability in the case of F1.

Table 12.
Results for damage scenario DS-13 using functions F1 and F4.

| Element | $\beta_{real}$ | $\beta_{comp}$ | |
|---|---|---|---|
| | | $F_1$ | $F_4$ |
| 3 | 0.26 | 0.26 (0) | 0.23 (12) |
| 8 | 0.22 | 0.22 (0) | 0.20 (9) |

Source: The authors.

Table 13.
Results for damage scenario DS-14 using functions F1 and F4.

| Element | $\beta_{real}$ | $\beta_{comp}$ | |
|---|---|---|---|
| | | $F_1$ | $F_4$ |
| 1 | --- | --- | 0.07 |
| 4 | 0.24 | 0.24 (0) | 0.21 (13) |
| 10 | 0.28 | 0.28 (0) | 0.17 (39) |
| 13 | 0.31 | 0.31 (0) | 0.25 (17) |
| 14 | --- | --- | 0.28 |

Source: The authors.

Table 14.
Results for damage scenario DS-15 using functions F1 and F4.

| Element | $\beta_{real}$ | $\beta_{comp}$ | |
|---|---|---|---|
| | | $F_1$ | $F_4$ |
| 2 | 0.25 | 0.26 (4) | 0.08 (68) |
| 10 | --- | --- | 0.17 |
| 13 | 0.25 | 0.25 (0) | 0.39 (56) |
| 15 | 0.25 | 0.26 (4) | 0.28 (12) |

Source: The authors.

Table 15.
Results for damage scenario DS-16 using functions F1 and F4.

| Element | $\beta_{real}$ | $\beta_{comp}$ | |
|---|---|---|---|
| | | $F_1$ | $F_4$ |
| 2 | --- | --- | 0.14 |
| 4 | --- | --- | 0.06 |
| 5 | 0.32 | 0.35(9) | 0.18 (44) |
| 11 | --- | --- | 0.07 |
| 26 | 0.29 | 0.29 (0) | 0.29 (0) |
| 35 | --- | --- | 0.05 |
| 38 | 0.24 | 0.24 (0) | 0.25 (4) |

Source: The authors.

Table 16.
Results for damage scenario DS-17 using functions F1 and F4.

| Element | $\beta_{real}$ | $\beta_{comp}$ | |
|---|---|---|---|
| | | $F_1$ | $F_4$ |
| 1 | 0.23 | 0.17 (26) | 0.24 (4) |
| 4 | --- | --- | 0.05 |
| 5 | --- | 0.20 | 0.05 |
| 7 | --- | --- | 0.05 |
| 11 | --- | --- | 0.09 |
| 12 | --- | --- | 0.07 |
| 13 | 0.31 | 0.32 (3) | 0.31(0) |
| 32 | 0.26 | 0.28 (8) | 0.23(12) |
| 41 | 0.18 | 0.20 (11) | 0.17 (6) |
| 43 | --- | --- | 0.06 |

Source: The authors.

The aforementioned result highlights the importance of decreasing computational error in terms of damage extent and misidentified elements quantity in order to enhance methodology performance.

Table 17.
Results for damage scenario DS-18 using functions F1 and F4.

| Element | $\beta_{real}$ | $\beta_{comp}$ | |
|---|---|---|---|
| | | $F_1$ | $F_4$ |
| 1 | --- | 0.09 | |
| 7 | 0.30 | 0.27 (10) | 0.28 (7) |
| 23 | 0.30 | 0.29 (3) | 0.29 (3) |
| 28 | --- | --- | 0.05 |
| 35 | 0.30 | 0.30 (0) | 0.31 (3) |
| 41 | 0.30 | 0.30 (0) | 0.30 (0) |

Source: The authors.

Table 18.
Results for damage scenario DS-19 using functions F1 and F4.

| Element | $\beta_{real}$ | $\beta_{comp}$ | |
|---|---|---|---|
| | | $F_1$ | $F_4$ |
| 3 | --- | --- | 0.05 |
| 4 | --- | --- | 0.06 |
| 13 | 0.19 | 0.20 (5) | 0.18 (5) |
| 15 | --- | --- | 0.13 |
| 22 | --- | 0.06 | --- |
| 25 | 0.24 | 0.24 (0) | 0.22 (8) |
| 28 | 0.20 | 0.19 (5) | 0.15 (25) |
| 29 | --- | 0.06 | --- |
| 30 | --- | 0.06 | --- |
| 34 | --- | 0.06 | --- |

Source: The authors.

Table 19.
Results for damage scenario DS-20 using functions F1 and F4.

| Element | $\beta_{real}$ | $\beta_{comp}$ | |
|---|---|---|---|
| | | $F_1$ | $F_4$ |
| 4 | --- | --- | 0.05 |
| 9 | 0.32 | 0.31(3) | 0.33 (3) |
| 11 | --- | --- | 0.05 |
| 14 | 0.28 | 0.27 (4) | 0.29 (4) |
| 15 | --- | --- | 0.06 |
| 27 | --- | --- | 0.05 |
| 30 | --- | 0.05 | --- |
| 34 | 0.22 | 0.25 (14) | 0.23(5) |
| 37 | 0.26 | 0.26 (0) | 0.21(19) |

Source: The authors.

Table 20.
Results for damage scenario DS-21 using functions F1 and F4.

| Element | $\beta_{real}$ | $\beta_{comp}$ | |
|---|---|---|---|
| | | $F_1$ | $F_4$ |
| 2 | 0.18 | 0.19 (0) | 0.18 (0) |
| 8 | --- | 0.05 | --- |
| 26 | --- | --- | 0.05 () |
| 27 | 0.18 | 0.19 (6) | 0.13 (28) |
| 28 | 0.18 | 0.18 (0) | 0.20 (11) |
| 29 | 0.18 | 0.17 (6) | 0.16 (11) |
| 45 | --- | --- | 0.05 |

Source: The authors.

## 6. Conclusions

This paper explores the application of the Differential Evolution (DE) to the damage detection problem. DE operators stem from the relevant scientific literature, with adaptive parameters based on the evolution of those parameters belonging to the best individual in the current

iteration. Such adaptation can be understood as one of the key contributions of this research as the final user need only define the population size. This methodology was tested on truss-type structures. Results show that the methodology detect the different damage scenarios (i.e. identifies both damaged elements and damage extent) with varying levels of reliability, depending on the objective function used. In this study, four objective functions were used, and the most reliable results were obtained by objective function F1, which is based on natural frequencies and mode shapes. Results for the function based on modal flexibility proved to be similar to those obtained with F1; even though the former generated more of misidentified elements. A different objective function, one based on natural frequencies and modal strain energies (F2), failed to locate the damaged element in most of the simple damage scenarios. As regards the objective function based on the residual force vector, the results were unreliable given that it led to numerous misidentified elements. As far as the ability to identify simple and multiple damage scenarios is concerned, the logical conclusion that simple damage scenarios are more reliably identified was confirmed. In conclusion, more research should be geared toward developing damage detection methodologies that find scenarios with many damaged elements.

## Acknowledgements

The authors would like to acknowledge CNPq (Brazilian National Council for Technological and Scientific Development) for the financial support provided to this research.

## References

[1] Talbi, E., Metaheuristics: From design to implementation. New Jersey: Ed. J. Wiley & Sons, 2009. http://dx.doi.org/10.1002/9780470496916

[2] Navarro, R., Puris, A. and Bello, R., The performance of some metaheuristics in continuous problems studied according to the location of the optima in the search space. DYNA, 80 (180), pp. 60-66, 2013.

[3] Begambre, O., Hybrid algorithm for damage detection: A heuristic approach, Ph. D. dissertation, University of Sao Paulo, Sao Carlos, SP, Brazil, 2007.

[4] Mares, C. and Surace, C., An application of genetic algorithm to identify damage in elastic structures. Journal of Sound and Vibration, 195 (2), pp. 95-215, 1996. http://dx.doi.org/10.1006/jsvi.1996.0416

[5] Moradi, S., Razi, P. and Fatahi, L., On the application of bees algorithm to the problem of crack detection of beam-type structures. Computers and Structures, 89, pp. 2169-2175, 2011. http://dx.doi.org/10.1016/j.compstruc.2011.08.020

[6] Fadel, L., Fadel, L.F. Kaminski, J. and Riera, J.D., Damage detection under ambient vibration by harmony search algorithm. Expert Systems with Applicatioon, 39, pp. 9704-9714, 2012. http://dx.doi.org/10.1016/j.eswa.2012.02.147

[7] Casciati, S., Stiffness identification and damage localization via differential evolution algorithms. Structural Control and Health Monitoring, 15, pp. 436-449, 2008. http://dx.doi.org/10.1002/stc.236

[8] Sandesh, K., Shankar, K., Application of a hybrid of particle swarm and genetic algorithm for structural damage detection. Inverse Problems in Science and Engineering, 18 (7), pp. 997-1021, 2012. http://dx.doi.org/10.1080/17415977.2010.500381

[9] Kyprianou, K. and Worden K., Identification of hysteretic systems sing the differential evolution algorithm. Journal of Sound and Vibration, 248 (2), pp. 289-314, 2001. http://dx.doi.org/10.1006/jsvi.2001.3798

[10] Patrick, H. Tinker, M. and Dozier, G., Evolutionary optimization of a geometrically refined truss. Structural Multidisciplinary. Optimization, 31, pp. 311-319, 2006. http://dx.doi.org/10.1007/s00158-005-0564-7

[11] Zhang, J. and Sanderson A.C., JADE: Adaptive differential evolution with optional external archive. IEEE Transactions on Evolutionary Computation, 13 (5), pp. 945-958, 2009. http://dx.doi.org/10.1109/TEVC.2009.2014613

[12] Brest, J. and Maucec, M.S., Self-adaptive differential evolution algorithm using population size reduction and three strategies. Soft Computing, 15, pp. 2157-2174, 2011. http://dx.doi.org/10.1007/s00500-010-0644-5

[13] Storn, R. and Price, K.V., Differential evolution- A simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012. Berkeley, USA: Intern. Computer Science Institute, 1995.

[14] Storn, R. and Price, K.V., Differential evolution- A simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization, 11 (4), pp. 341-359, 1997. http://dx.doi.org/10.1023/A:1008202821328

[15] Tvridk, J., Differential evolution: competitive setting of control parameters. Proceedings of the International Multiconference on Computer Science and Information Technology, pp. 207-213, 2006.

[16] Ali, M.M. and Torn, A., Population set-based global optimization algorithms: some modifications and numerical studies. Computers and Operation. Research, 31, pp. 1703-1725, 2004. http://dx.doi.org/10.1016/S0305-0548(03)00116-3

[17] Villalba, J.D. and Laier, J.E., Localising and quantifying damage by means of a multi-chromosome genetic algorithm. Advances in Engineering Software, 50, pp. 150-157, 2012. http://dx.doi.org/10.1016/j.advengsoft.2012.02.002

[18] Perera, R. and Torres, R., Structural damage detection via modal data with genetic algorithms. Journal of Structural Engineering, 132 (93), pp. 1491-1501, 2006. http://dx.doi.org/10.1061/(ASCE)0733-9445(2006)132:9(1491)

[19] Raich, A. and Liszkai, T., Improving the performance of structural damage detection methods using advanced genetic algorithms. Journal of Structural Engineering, 133 (3), pp. 449-461, 2007. http://dx.doi.org/10.1061/(ASCE)0733-9445(2007)133:3(449)

**J.D. Villalba-Morales,** received his BSc. Eng in Civil Engineering in 2005 at the Industrial University of Santander, Bucaramanga, Colombia. He obtained his MSc and Dr.in 2007 and 2012 at the University of Sao Paulo, Brazil. After Sao Paulo, Dr. Villalba began to work as an assistant professor in the Pontificia Universidad Javeriana, Bogotá, Colombia, from 2013 to the present, where he teaches courses on numerical methods for engineers as well as the finite element method. His research interests include structural health monitoring, earthquake engineering, steel structures, optimization and computational intelligence technique application and development.

**J.E. Laier,** received his BSc. Eng in Civil Engineering in 1971 at the University of Sao Paulo, Brazil, the university from which he also obtained his MSc. In 1975 and Dr. in Structural Engineering in 1978. He continued research as a PhD at The Computational Mechanics Center, Southampton, UK. Currently, he teaches as a full professor at the University of Sao Paulo, Brasil. His experience in the Civil Engineering field, focused on structural mechanics, includes work on numerical analysis, structural dynamics, high-rise building analysis, damage detection and structural identification.