2012

Localising and quantifying damage by means of a multi-chromosome genetic algorithm

ADVANCES IN ENGINEERING SOFTWARE, OXFORD, v. 50, n. 47, supl. 1, Part 1, pp. 150-157, AUG, 2012
http://www.producao.usp.br/handle/BDPI/41335

Advances in Engineering Software 50 (2012) 150–157

Contents lists available at SciVerse ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Localising and quantifying damage by means of a multi-chromosome genetic algorithm

J.D. Villalba *, J.E. Laier

Department of Structural Engineering, University of Sao Paulo, Av. Trabalhador Saocarlense 400, CEP 13566-590, São Carlos, Brazil

ARTICLE INFO

Article history:
Available online 10 March 2012

Keywords:
Dynamic parameters
Damage detection
Genetic algorithms
Truss structures
Self-adaptation
Multi-chromosome

ABSTRACT

This paper presents a structural damage detection methodology based on genetic algorithms and dynamic parameters. Three chromosomes are used to codify an individual in the population. The first and second chromosomes locate and quantify damage, respectively. The third permits the self-adaptation of the genetic parameters. The natural frequencies and mode shapes are used to formulate the objective function. A numerical analysis was performed for several truss structures under different damage scenarios. The results have shown that the methodology can reliably identify damage scenarios using noisy measurements and that it results in only a few misidentified elements.

1. Introduction

Different factors, like aging, fatigue or exposure to aggressive environments, can damage a structure. One of the principal ways to detect damage is by considering the changes that occur in the dynamical parameters after damage (e.g., natural frequencies, mode shapes and modal damping). These changes reflect variations in the structural properties (stiffness, damping and mass matrices), permitting an optimisation problem to be formulated. The idea is to minimise the differences between the dynamic parameters obtained in an experimental test and those obtained from a finite element model that represents the damaged structure. This model is updated from another finite element model that defines the undamaged state of the structure. A complete review of damage detection methodologies based on dynamic parameters can be found in [1–3].

A genetic algorithm is an optimisation technique that can be used to solve the structural damage detection problem. Moslem and Nafaspour [4] used a technique based on a residual force vector to define the probably damaged elements with the advantage of reducing the effects of noise in the measurements. A steady-state genetic algorithm was implemented to compute the damage extent. Ananda Rao et al. [5] proposed a simple genetic algorithm with binary representation and an objective function that is based on the residual force vector. This type of function has the disadvantage of requiring complete mode shapes. He and Hwang [6] combined a real-coded genetic algorithm with a simulated annealing algorithm to detect damage in beam structures. Natural frequencies and displacements of static response were used to generate the objective function. Raich and Liszkai [7] used changes in the frequency response functions and an implicit redundant representation. This type of representation permits the dynamic variation of the variables to be optimised during the evolutionary process. The above characteristic is important in the solution to the damage detection problem, as the number and localisation of the damaged elements are not known *a priori*. Kouchmeshky et al. [8] detected damage in an iterative process by applying two phases: estimation and exploration. In the first phase, they used an objective function that was based on frequency response functions and represented a possible solution by using one chromosome to define the location of the damage and another to determine the damage extent. The second phase was applied to the candidate solutions found in the first phase. Meruane and Heylen [9] proposed a parallel real-coded genetic algorithm based on migration to quantify damage in structures. The objective function was characterised by permitting the use of operational modes, by producing a low quantity of misidentified elements and, also, by taking into account errors in the finite-element model for the initial condition.

This paper presents an improved version of the damage detection methodology that is based on genetic algorithms and dynamic parameters, which was presented by the authors at the Tenth International Conference on Computational Structures Technology [10]. The new contributions are related to the generation of a self-adaptive algorithm and to the formulation of a new objective

* Corresponding author.
E-mail address: villalba@sc.usp.br (J.D. Villalba).

0965-9978/$ - see front matter
doi:10.1016/j.advengsoft.2012.02.002

function, which produces more reliable results than the previous one.

2. Damage representation

The undamaged and damaged conditions are represented by finite element models with the current model obtained by updating the undamaged model. In this research, the damage was considered a reduction in the elasticity modulus of the damaged element [11]. The relationship between the two conditions for the jth element can be given by

$$E_{dj} = (1 - \beta_j) \times E_j, \tag{1}$$

where E_{dj} and E_j are the elasticity moduli of element j for the damaged and undamaged conditions, respectively. β_j is an elasticity modulus reduction factor with a value equal to 0 if the element is not damaged, and a value equal to 1 for the completely damaged state. This factor must be computed for each element in the structure and corresponds to the variable to be optimised by the genetic algorithm.

3. Genetic algorithms

Genetic algorithms are analogous to the natural selection laws and survival of strongest individual [12]. They are used to derive a set of possible solutions over the search space and to find an answer to a specific problem. First, the implementation of an algorithm requires the definition of two important aspects: a codification scheme for the possible solutions and the definition of an objective function. Next, it is necessary to specify the population size to be used and to determinate if the initial population will be generated in a random or heuristic way. Selection, crossover and mutation are applied to the current population. In the selection process, the best individuals in the population are chosen according to their fitness level. The fitness is computed by using the objective function defined for the optimisation problem. The crossover process generates two new solutions from two other solutions chosen in the selection process. The mutation process consists of introducing random variations in the new individuals. The last two operators are limited by the rates that define their probability of being applied. Elitism is an extra process that can be applied, and it works by copying the best individual from the past generation and placing it in the next generation. The above steps are performed iteratively until either obtaining convergence or reaching a pre-specified number of generations. Due to the characteristics of the genetic algorithms, several executions are often necessary to find a final answer to the problem. For a complete reference of genetic algorithms, the reader should see [12] or [13].

The formulation for the classic genetic algorithms can be found in the above texts; however, many advanced algorithms have been proposed in the literature. One of them is the multi-chromosome genetic algorithm, which uses several chromosomes to generate solutions for complex problems. Hinterding [14] solved the cutting stock problem by using one chromosome to stand for a possible solution and another to self-adapt the genetic parameters. Baine [15] used four chromosomes to represent the input and output fuzzy sets of a proportional-plus-derivative fuzzy logic controller. Király and Abonyi [16] codified the possible solutions to the multiple travelling salesmen problem by assigning one chromosome for each salesman in the solution.

On the other hand, the setting of genetic parameters can be a long process because the selection of values for the population size, mutation and crossover rates should be related directly to the studied problem. Currently, an important research topic is the development of strategies to adapt these parameters [17,18].

4. Damage detection methodology

The proposed damage detection methodology consists of solving an optimisation problem through an objective function based on dynamic parameters. The solution to the optimisation problem was performed by applying the steps shown in Fig. 1.

- STEP 1: Define the finite element model for the undamaged structure.

A finite element model is required to represent the undamaged condition for the structure and to obtain a new model for the damaged condition, which is performed in an updating process. The stiffness, K_{str}, and mass, M_{str}, matrices are computed from the contribution of each element in the structure and are given as

$$K_{str} = \sum_{i=1}^{nelem} k_i, \tag{2}$$

$$M_{str} = \sum_{i=1}^{nelem} m_i, \tag{3}$$

where k_i and m_i are the stiffness and mass matrices for the ith element, respectively, and *nelem* is the number of elements in the structure. In this paper, the structure is considered undamped.

- STEP 2: Assess the experimental dynamic parameters of the damaged structure.

The dynamic parameters for the current structure (natural frequencies and mode shapes) have to be experimentally determined. This research was carried out numerically, so the damaged dynamic parameters were computed by introducing the damage scenario into the undamaged stiffness matrix and then solving the eigen-problem, which is expressed by the following equation:

$$(K_{str,d} - \omega_d^2 M_{str})\phi_d = 0, \tag{4}$$

where ω is the natural frequency and ϕ is the mode shape. The subscript d refers to the damaged condition.

The damaged stiffness matrix is computed as a function of the undamaged stiffness matrix, as follows:

$$K_{str,d} = K_{est} - \sum_{i=1}^{den} k(\beta_i E_i)_i, \tag{5}$$

where *den* is the number of damaged elements in the analysed damage scenario.

In a real dynamic test, it is impossible to avoid the presence of noise; therefore this fact was simulated through the perturbation of the computed damaged dynamic parameters. The following equations permit simulating the noise in the measurements:

$$\omega_{dr} = \omega_d \times (1 + \text{random}(-1,1) \times \text{Noise}_f), \tag{6}$$

$$\phi_{ijdr} = \phi_{i,jd} \times (1 + \text{random}(-1,1) \times \text{Noise}_\phi), \tag{7}$$

Begin

1. Define the finite element model for the undamaged structure.
2. Assess the experimental dynamic parameters of the damaged structure.
3. Formulate the objective function.
4. Configure the genetic algorithm.
5. Execute the algorithm and show the damage scenario found.

End

Fig. 1. Damage detection methodology.

where *dr* denotes a noisy value. $Noise_f = 1\%$ and $Noise_\phi = 3\%$ are the perturbations of the natural frequencies and mode shapes, respectively [19].

- STEP 3: Formulate the objective function.

In this step, a maximisation problem was solved by using an objective function that is based on the natural frequencies and mode shapes. The objective function is given by

$$G = \sum_{j=1}^{nm} \frac{c_1}{c_2 + F_j}, \tag{8}$$

with

$$F_j = \left| \frac{\omega_j^{ga} - \omega_j^{ex}}{\omega_j^{ex}} \right| + W \times \sqrt{\frac{\sum_{i=1}^{ndf} (\phi_{ij}^{ga} - \phi_{ij}^{ex})^2}{\sum_{i=1}^{ndf} (\phi_{ij}^{ex})^2}}, \tag{9}$$

where *nm* refers to the number of vibration modes considered and *ndf* is the number of degrees of freedom with the available information. The superscript *ga* refers to results from the finite element model obtained by the genetic algorithm and the superscript *ex* indicates the experimental values. ω_j is the *j*th natural frequency and $\phi_{i,j}$ is the magnitude of the *j*th mode shape for the *i*th degree of freedom. c_1 and c_2 are constants defined by the user and herein they take values of 200 and 1, respectively. Constant c_1 limits the value of the objective function, while constant c_2 avoids the division by zero when the data are free of noise. The weight factor $W = 2.0$ was defined after several trials. This type of definition for the weight factor can be avoided by using multi-objective formulations [20]. This formulation will be studied by the authors in future work.

- STEP 4: Configure the genetic algorithm.

One of the principal difficulties in solving the damage detection problem is that the number and position of the real damaged elements are unknown at the beginning of the optimisation process. Therefore, we used a type of representation with multiple chromosomes, which allows the number of damaged elements in an individual to change throughout the evolutionary process. An individual with three chromosomes was proposed by following the ideas presented above in [8,14]. The first chromosome is a real-type one and represents the damage extent, which can range between 0 and 1. The second is a binary-type and is used to locate the damage. A value of one identifies a damaged element. Each gene corresponds to one element in the structure, so a structure with N elements will have N genes for the above chromosomes. The third chromosome is a real-type one and is employed to allow self-adaptation of the genetic parameters. Fig. 2 shows one of the possible individuals for a structure with six elements. In this damage scenario, elements 1, 3 and 4 have damage values of 0.25, 0.18 and 0.35, respectively. The third chromosome codifies, from left to right, the binary and real crossover rates and the binary and real mutation rates. In this way, the user does not precise determine an optimal configuration for the genetic parameters. They are considered optimisation variables and their optimum values will be determined throughout the evolutionary process. The genetic algorithm configured here was called self-adaptive multi-chromosome genetic algorithm (SAMGA).

Real Chromosome - Damage Quantification

0.25	0.07	0.18	0.35	0.20	0.19

Binary Chromosome - Damage Localization

1	0	1	1	0	0

Real Chromosome - Genetic Parameters

0.85	0.90	0.01	0.04

Fig. 2. Typical chromosomes for representing an individual.

Table 1
The limits of the genetic parameters for the simulation.

Boundary	Binary crossover	Real crossover	Binary mutation	Real mutation
Lower	0.70	0.80	0.005	0.03
Upper	0.90	0.95	0.02	0.06

Table 2
The multi-chromosome genetic algorithm operators.

Chromosome	Selection	Crossover	Mutation
Real	Tournament, $n = 3$	BlX-α, $\alpha = 0.5$	Creep
Binary		Two points	Jump

Shown in Fig. 2, two different individuals in the population may represent damage scenarios with different numbers of damaged elements. But, at the end of the generations, the best individual in the population is expected to correspond to the real set of damaged elements and the correct damage level.

The initial population was generated heuristically, considering that only a few elements could be damaged, the damage extent was not severe and the genetic parameters could assume a range of allowed values. Each gene in the first chromosome assumed a random value between 0 and 0.5 and a random value of either 0 or 1, for the genes in the second chromosome. The range for each operator rate (Table 1) was chosen based on our experience [10,21,22] and some trials.

Table 2 shows the genetic operators applied to each chromosome type, and the way they are applied can be found in [23,24]. In that table n is the number of participants in the tournament and α is a parameter that influences the crossover between individuals. The value chosen for the latter parameter determines the balance between the exploration and the exploitation of the search space [24]. In the crossover process, we used the crossover rates from the parent individual with the highest fitness. The binary and real mutation rates that were used to mutate a specific gene corresponded to the values of the individuals obtained after the crossover process. The only parameter to be defined is the population size, whose criterion of choice will be discussed in the following section.

- STEP 5: Execute the algorithm and show the damage scenario found.

Ten executions are performed and the solution with the best fitness is chosen as the damage scenario found. Satisfying one of the following criteria is used to stop the execution of the algorithm: (1) a maximum number of generations equal to 400 or (2) a predefined number of consecutive generations without a significant change in the fitness of the best individual, which is 50 generations in this case.

5. Numerical examples

5.1. Analysed structures and damage scenarios

Several truss structures were analysed (Fig. 3). All of the elements in the analysed structures have an elasticity modulus $E = 200 \times 10^9$ N/m^2, density $\rho = 7800$ kg/m^3, and cross-sectional area $A = 0.001$ m^2. The six first modes were considered to be known for all of the structures. The maximum number of degree of freedom for the analysed structures was 47.

One genetic parameter, i.e. the population size, remains to undefined. Generally, the value for this parameter is obtained after

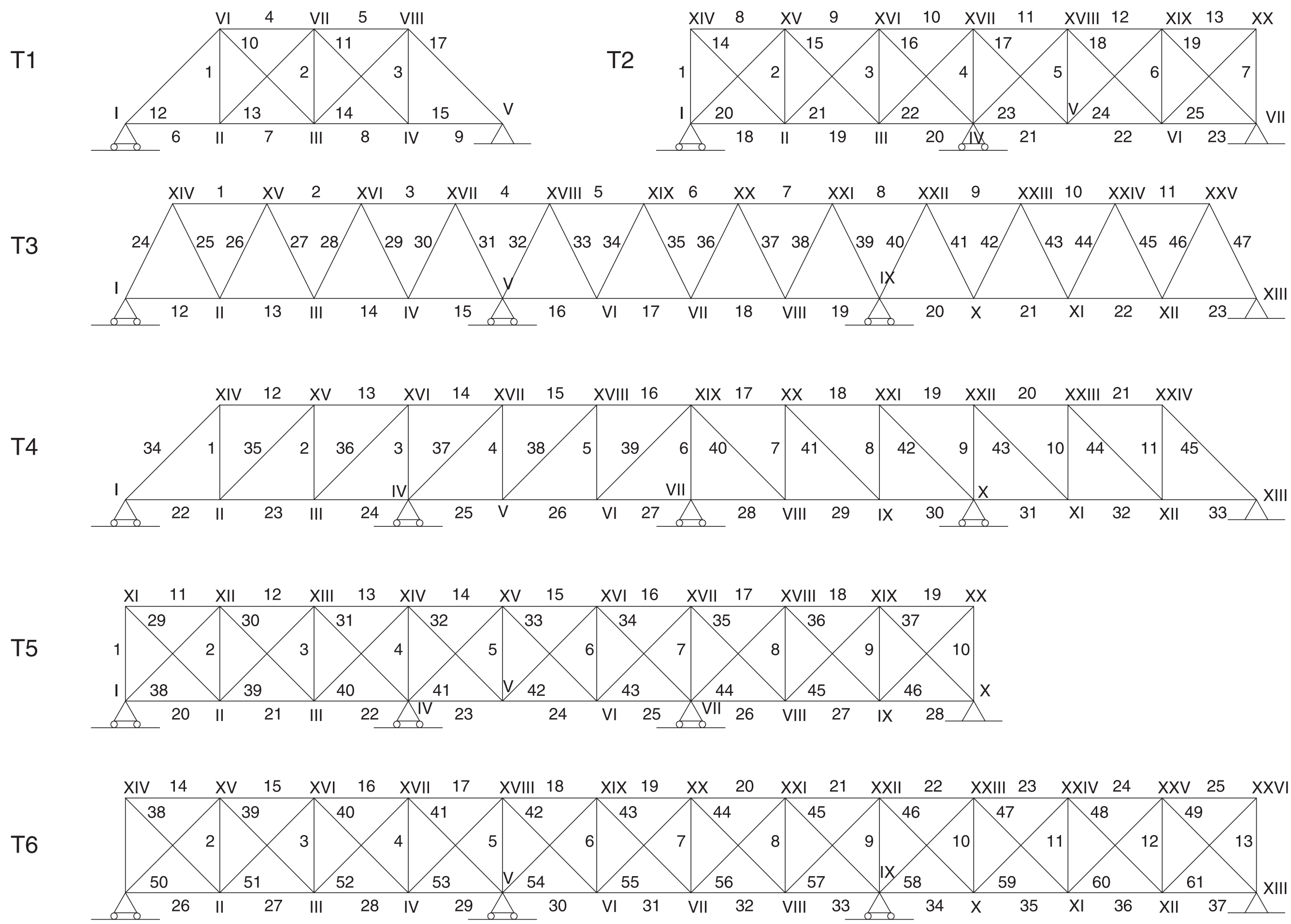

Fig. 3. The structures of the analysed trusses.

Table 3
Simple damage scenarios.

Structure	ID scenario	Damaged element	Damage β
T1	S1	1	0.310
	S2	7	0.180
	S3	13	0.330
T2	S4	3	0.480
	S5	9	0.350
	S6	27	0.240
T3	S7	1	0.290
	S8	29	0.250
	S9	30	0.450
T4	S10	5	0.480
	S11	26	0.180
	S12	38	0.410
T5	S13	8	0.220
	S14	17	0.450
	S15	35	0.280
T6	S16	3	0.330
	S17	31	0.300
	S18	44	0.190

some trials that consist in varying the population size until one obtains a desired performance of the proposed methodology. Ref. [4] presents one of the few heuristics that have been proposed to compute the size population as a function of the analysed structure. In that work the population size was defined to be directly proportional to the number of element in the structure. Based on our experience, we defined the population size as 200 for all of the simulated examples. This value is adequate for the type and sizes of the analysed structures, but it may not be enough to guarantee a good performance of the methodology when larger truss structures are analysed.

Five damage scenarios are simulated for each structure. These scenarios can be classified into simple and multiple damage scenarios as a function of the number of damaged elements (Tables 3 and 4).

6. Results

Tables 5–10 show the results found by the proposed methodology for the different damage scenarios. In these tables, only the damaged elements that were identified by the binary chromosome and have a damage extent higher than 0.03 are shown. The real damage scenarios are shown and the misidentified elements found by the proposed methodology are underlined for a better understanding of the results.

Table 5 shows the results for the simulated damages scenarios in structure T1. All of the damage scenarios were correctly found. Only one misidentified element was computed for damage scenario S2 with a low damage extent. The maximum error in the computation of the damage extent was less than 0.009 and a minimum error of 0.002 was found for scenario S1. A smaller population size could have been used for this structure, but the role of the population size was not analysed in this paper.

The performance of our methodology to detect the damage scenarios in structure T2 is shown in Table 6. Nearly all of the

Table 4
Multiple damage scenarios.

Structure	ID scenario	Damaged element	Damage β
T1	M1	1	0.470
		7	0.250
		13	0.300
	M2	2	0.160
		6	0.200
		11	0.200
T2	M3	3	0.330
		9	0.230
		27	0.260
	M4	2	0.420
		22	0.300
		30	0.350
T3	M5	1	0.270
		29	0.230
		30	0.260
	M6	8	0.150
		9	0.150
		33	0.150
T4	M7	5	0.160
		26	0.200
		38	0.180
	M8	7	0.340
		35	0.200
		36	0.250
T5	M9	8	0.400
		17	0.350
		35	0.450
	M10	30	0.380
		5	0.280
		24	0.300
T6	M11	30	0.150
		31	0.180
		44	0.150
	M12	8	0.200
		39	0.240
		42	0.270

damage scenarios resulted in at least one misidentified element with damage less than 0.07. The errors in the computed damage for the real damaged elements were less than 0.012. Identical damage scenarios were simulated in the absence of noise. The exact damage scenario was found for all of the examples, which indicates that more reliable results can be obtained if we diminish the noise level in the measurements. Also, the methodology was tested considering measurements with high levels of noise, until 2% in natural frequencies and 10% in mode shapes. It was observed that the real damaged elements were identified and the number of misidentified elements increased as the noise increased.

Table 7 shows the results for the different damage scenarios that were applied to structure T3. The error in the computation of the damage extent was less than 0.015 in all cases. More than one misidentified element was computed for damage scenarios S7, M5 and M6. These elements generally had low values of damage, but element 26 in scenario M6 showed a high damage value. However, the performance of the proposed methodology can be considered satisfactory.

The results for structure T4 are shown in Table 8. All of the simple damage scenarios had some misidentified elements. The methodology did not have a good performance in the case of multiple damage scenarios, as element 38 in scenario M7 was not identified as damaged. In a detailed analysis, it was observed that the methodology found the real set of damaged elements in three of the ten executions. These executions presented solutions with a slightly lower fitness than that of the best execution. This fact would indicate that the method used to select the final solution might not work correctly if the values of the objective function for one or more damage scenarios were similar to that of the correct solution. This issue could be resolved by one of the following measures: (1) explore another objective function that more reliably differentiates between solutions, (2) use a strategy to find multiple optima or (3) propose a new scheme to better define the final solution.

Table 9 shows the results for the damage scenarios in structure T5. Damage scenario S14 had four misidentified elements; which is a small percentage of the total number of elements in the structure. The damage extent for those elements was not less than 0.08, which is a relatively low damage value. The error associated with the assessment of damage in the real damaged elements was less than 0.035.

The damage scenarios analysed for structure T6, simple and multiple, produced the greatest number of misidentified elements (Table 10). However, all of the real damaged elements were identified with differences in the value of the damage less than 0.033. The best performance was obtained in the identification of damage scenario S18 in which a horizontal element was damaged. For this scenario, only one element was misidentified and the damage value was low.

In general, it is observed that simple damage scenarios can be found with more confidence than multiple damage scenarios. The difference between the real and computed damage values was less than 0.05, which may have originated from the presence of noise in the measurements. Most of the cases produced a few misidentified elements with low values of damage. However, it was possible to observe some scenarios that had misidentified elements with a damage values greater than 0.1, such as scenarios M6 and S10. The performance of the damage detection methodology may depend on the damage scenario analysed, therefore it is not possible to guarantee that the methodology works in 100% of the cases.

6.1. Convergence, generations and misidentified elements

The number of correct executions out of ten, the average number of misidentified elements and the average number of generations for each damage scenario simulated are shown in Table 11. The performance of the methodology is excellent at detecting simple damage scenarios, producing only a few misidentified elements

Table 5
The results for structure T1.

	ID scenario									
	S1		S2		S3		M1		M2	
	Elem.	β_j	Elem.	β_j	Elem.	β_j	Elem.	β_j	Elem.	β_j
Real	1	0.310	7	0.180	13	0.330	1	0.470	2	0.160
							7	0.250	6	0.200
							13	0.300	11	0.200
GA	1	0.308	7	0.174	13	0.333	1	0.472	2	0.168
			8	0.039			7	0.253	6	0.200
							13	0.309	11	0.203

Table 6
The results for structure T2.

	ID scenario									
	S4		S5		S6		M3		M4	
	Elem.	β_j	Elem.	β_j	Elem.	β_j	Elem.	β_j	Elem.	β_j
Real	3	0.480	9	0.350	27	0.240	3	0.330	2	0.420
							9	0.230	22	0.300
							27	0.260	30	0.350
GA	3	0.475	9	0.346	27	0.239	3	0.331	2	0.386
	18	0.068	18	0.041			6	0.046	18	0.075
							9	0.229	22	0.312
							27	0.258	30	0.344

Table 7
The results for structure T3.

	ID Scenario									
	S7		S8		S9		M5		M6	
	Elem.	β_j	Elem.	β_j	Elem.	β_j	Elem.	β_j	Elem.	β_j
Real	1	0.290	29	0.250	30	0.450	1	0.270	8	0.150
							29	0.230	9	0.150
							30	0.260	33	0.150
GA	1	0.290	26	0.039	30	0.448	1	0.274	8	0.160
	9	0.036	29	0.262			12	0.042	9	0.146
	30	0.042					29	0.269	10	0.034
	37	0.061					30	0.443	26	0.229
	42	0.061					34	0.032	33	0.181

Table 8
The results for structure T4.

	ID scenario									
	S10		S11		S12		M7		M8	
	Elem.	β_j	Elem.	β_j	Elem.	β_j	Elem.	β_j	Elem.	β_j
Real	5	0.480	26	0.180	38	0.410	5	0.160	7	0.340
							26	0.200	35	0.200
							38	0.180	36	0.250
GA	5	0.480	26	0.173	1	0.140	1	0.127	1	0.064
	19	0.161	31	0.033	5	0.069	5	0.242	7	0.327
	21	0.029			8	0.117	10	0.127	19	0.082
	44	0.044			12	0.042	26	0.188	35	0.114
					21	0.075	30	0.043	36	0.248
					32	0.036	32	0.032	38	0.092
					35	0.086				
					38	0.448				
					44	0.093				

and reaching the convergence to the solution in a few generations. For multiple damage scenarios, the methodology may not find the total of real damaged elements in all the executions, as it can be observed in the results for the M7 scenario. An average of seven misidentified elements was found, but this quantity corresponded to only 16% of the elements in the structure. Convergence was reached before completing the maximum number of generations for both types of damage scenarios. Therefore, it can be concluded that different damage scenarios will be found with different reliability levels.

6.2. Evolution of the number of damaged elements

Figs. 4 and 5 show how the Number of Damaged Elements in the binary chromosome of the best Individual (NDEI) decreases with the generation number. This is the principal characteristic of the proposed genetic algorithm because it permits convergence to a solution with a few damaged elements. The number of elements may increase between consecutive generations, but the overall trend is to decrease. For scenario S9, four damaged elements were found, but Table 3 shows only one element because the other three elements had damage values of less than 0.03. Damage scenario M5 had three damaged elements and our method detected seven damaged elements. Some of the misidentified elements had damage values greater than 0.03.

6.3. Evolution in the genetic parameters

Fig. 6 shows the self-adaptation of the average genetic parameters for two damage scenarios in structure T4. These parameters did not converge to a specific value before the algorithm converges to the solution of the problem. A similar behaviour was observed for the other damage scenarios. Therefore, it should be pointed

Table 9
The results for structure T5.

	ID Scenario									
	S13		S14		S15		M9		M10	
	Elem.	β_j	Elem.	β_j	Elem.	β_j	Elem.	β_j	Elem.	β_j
Real	8	0.220	17	0.450	35	0.280	8	0.400	3	0.380
							17	0.350	5	0.280
							35	0.450	24	0.300
GA	8	0.185	9	0.072	6	0.065	6	0.110	3	0.378
	33	0.037	17	0.446	19	0.046	8	0.377	5	0.273
			27	0.047	27	0.037	17	0.358	24	0.320
			29	0.063	35	0.256	33	0.049	27	0.032
			30	0.034			35	0.438		

Table 10
The results for structure T6.

	ID Scenario									
	S16		S17		S18		M11		M12	
	Elem.	β_j	Elem.	β_j	Elem.	β_j	Elem.	β_j	Elem.	β_j
Real	3	0.330	31	0.300	44	0.190	3	0.150	8	0.200
							31	0.180	39	0.240
							44	0.150	42	0.270
GA	3	0.322	3	0.126	25	0.055	3	0.125	7	0.102
	11	0.138	7	0.109	44	0.184	8	0.042	8	0.204
	12	0.073	8	0.087			11	0.085	21	0.060
	20	0.032	11	0.164			25	0.071	39	0.211
	22	0.041	31	0.298			26	0.071	42	0.303
	25	0.071	47	0.125			31	0.165	50	0.033
	31	0.043	48	0.037			44	0.141		
	39	0.029	49	0.029			56	0.092		
			53	0.046			57	0.038		

Table 11
Performance of the SAMGA for the different damage scenarios.

ID scenarios	Correct runs	Misidentified elements	Generations
S1	10	3	87
S2	10	3	91
S3	10	2	87
S4	10	3	115
S5	10	2	92
S6	10	3	118
S7	10	6	150
S8	10	3	120
S9	10	3	145
S10	10	11	354
S11	9	10	343
S12	10	13	387
S13	10	4	137
S14	10	5	126
S15	10	4	127
S16	10	10	144
S17	10	8	138
S18	10	4	117
M1	10	3	72
M2	10	4	114
M3	10	6	197
M4	10	6	192
M5	9	6	173
M6	10	8	197
M7	3	12	383
M8	3	7	262
M9	10	7	170
M10	10	5	108
M11	9	8	184
M12	10	7	157

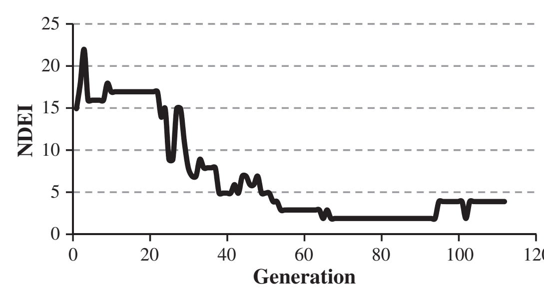

Fig. 4. Scenario S9: one damaged element.

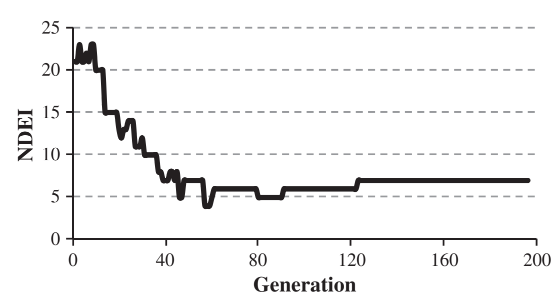

Fig. 5. Scenario M5: three damaged elements.

out that the values of the genetic parameters seem to depend on each specific run. On the other hand, the principal contribution of the proposed methodology is that it eliminates the need to

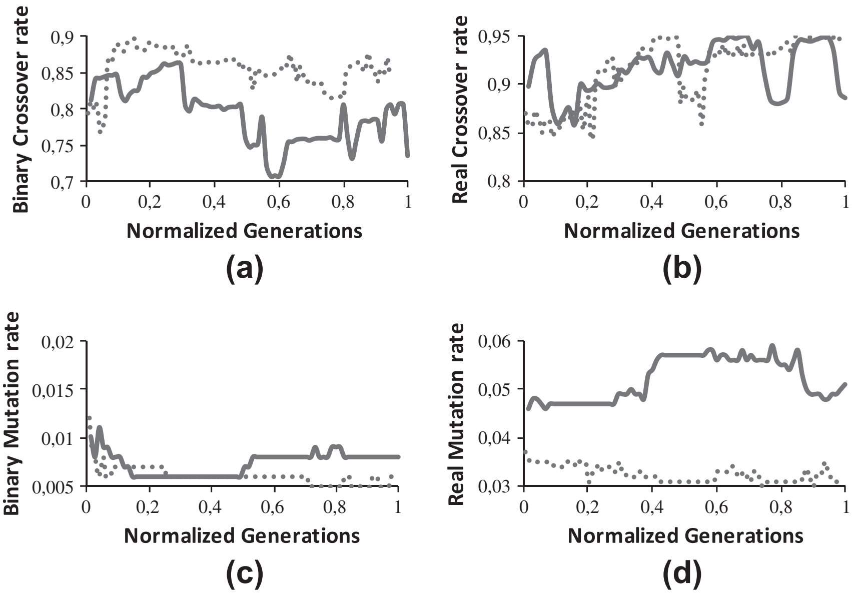

Fig. 6. The evolution of the genetic parameters: (a) binary crossover rate, (b) real crossover rate, (c) binary mutation rate and (d) real mutation rate. Scenario M8: solid line. Scenario M7: points sequence.

provide initial values for the genetic parameters. More studies must be performed to propose better self-adaptation techniques.

7. Conclusions

This paper proposes the use of a self-adaptive multi-chromosome genetic algorithm (SAMGA) to locate and quantify damage in structures. This type of algorithm permits representing the damage extent and the position of the damage by using two different chromosomes. A third chromosome is used for the self-adaptation of the genetic parameters, but the population size must be defined by trials. The results show that the proposed methodology can reliably determine the real damaged elements and the damage extent for different damage scenarios. Among all of the damage scenarios analysed, only one damaged element was not found and a few elements were misidentified as damaged elements. These elements generally presented a low value of damage. The success of this algorithm can be attributed to the fact that the number of damaged elements in each individual of the population can change during the evolutionary process.

Acknowledgements

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

References

[1] Doebling SW, Farrar CR, Prime MB. A summary review of vibration-based damage identification methods. Shock Vib Dig 1998;30(2):91–105.
[2] Carden EP, Fanning P. Vibration based condition monitoring: a review. Struct Health Monit 2004;3(4):355–77.
[3] Fan W, Qiao P. Vibration-based damage identification methods: a review and comparative study. Struct Health Monit 2011;10:83–111.
[4] Moslem K, Nafaspour R. Structural damage detection by genetic algorithms. AIAA J 2002;40(7):1395–401.
[5] Ananda Rao M, Srinivas J, Murthy BSN. Damage detection in vibrating bodies using genetic algorithms. Comput Struct 2004;82:963–8.
[6] He RS, Hwang SF. Damage detection by an adaptive real-parameter simulated annealing genetic algorithm. Comput Struct 2006;84:2231–43.
[7] Raich A, Liszkai T. Improving the performance of structural damage detection methods using advanced genetic algorithms. J Struct Eng 2007;133(3):449–61.
[8] Kouchmeshky B, Aquino W, Billek A. Structural damage identification using co-evolution and frequency response functions. Struct Control Health Monit 2008;15:162–82.
[9] Meruane V, Heylen W. Damage detection with parallel genetic algorithms and operational modes. Struct Health Monit 2010;9(6):481–96.
[10] Villalba JD, Laier JE. Advanced genetic algorithm for structural damage detection. In: Topping BHV, Adam JM, Pallares FJ, Bru R, Romero ML, editors, Proceedings of the tenth international conference on computational structures technology. Stirlingshire, Scotland: Civil-Comp Press, paper 44; 2010.
[11] Koh BH, Dyke SJ. Structural health monitoring for flexible bridge structures using correlation and sensitivity of modal data. Comput Struct 2007;85:117–30.
[12] Goldberg D. Genetic algorithms in search optimization, and machine learning. Reading, MA: Addison-Wesley Publishing Company; 1989.
[13] Michalewicz Z. Genetic algorithms + data structures = evolution programs. 2nd ed. Berlin: Springer-Verlag; 1994.
[14] Hinterding R. Self-adaptation using multi-chromosomes. In: Proceedings of the IEEE international conference on evolutionary computation, Indianapolis, United States; 1997. p. 87–91.
[15] Baine N. A simple multi-chromosome genetic algorithm optimization of a proportional-plus-derivative fuzzy logic controller. In: Proceedings of the annual meeting of the north american fuzzy information processing society, IEEE Press, New York City, United States; 2008, p. 1–5.
[16] Király A, Abonyi J. Optimization of multiple traveling salesmen problem by a novel representation-based genetic algorithm. In: Tenth international symposium of Hungarian researchers on computational intelligence and informatics, Budapest, Hungary; 2009. p. 315–26.
[17] Meyer-Nieberg S, Beyer HG. Self-adaptation in evolutionary algorithms. In: Lobo Fernando G, Lima Cláudio F, Michalewicz Zbigniew, editors. Parameter setting in evolutionary algorithms. Heidelberg: Springer Verlag; 2007. p. 47–76.
[18] Eiben AE, Michalewicz Z, Schoenauer M, Smith JE. Parameter control in evolutionary algorithms. In: Lobo Fernando G, Lima Cláudio F, Michalewicz Zbigniew, editors. Parameter setting in evolutionary algorithms. Heidelberg: Springer Verlag; 2007. p. 19–46.
[19] Chen B, Nagarajaiah S. Flexibility-based structural damage identification using Gauss- Newton method. In: Proceedings of SPIE – sensors and smart structures, technologies for civil, mechanical, and aerospace systems, San Diego, USA, vol. 6529 Part 1, paper 65291L; 2007.
[20] Perera R, Ruiz A, Manzano C. Performance assessment of multicriteria damage identification genetic algorithms. Comput Struct 2009;87:120–7.
[21] Laier JE, Morales JDV. Improved genetic algorithm for structural damage detection. In: Yong Yuan Y, Cui J, Mang, editors, Computational structural engineering: proceedings of the international symposium on computational structural engineering, Shanghai, China, part 8; 2009. p. 833–839.
[22] Villalba JD, Laier JE. Two-step genetic algorithm methodology for structural damage detection by using parameters dynamics. In: Aguiar AR, editor, Proceedings of the 11th Pan-American congress of applied mechanics, Foz do iguaçu, Brazil, paper PAC-0081; 2010.
[23] Haupt R, Haupt S. Practical genetic algorithms. 2nd ed. New Jersey: John Wiley & Sons; 2004.
[24] Herrera F, Lozano M, Verdergay JL. Tackling real-coded genetic algorithms: operators and tools for behavioral analysis. Artif Intell Rev 1998;12:265–319.