INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng (in press)
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.1803

Co-evolutionary algorithm for structural damage identification using minimal physical testing

B. Kouchmeshky[1,‡], W. Aquino[1,*,†], J. C. Bongard[2,§] and H. Lipson[2,¶]

[1]*School of Civil and Environmental Engineering, Cornell University, Ithaca, NY 14853, U.S.A.*
[2]*Computational Synthesis Laboratory, Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, U.S.A.*

SUMMARY

The problem of damage identification using minimum test data is studied in this work. Data sparsity in damage identification applications commonly results in inverse problems that are mathematically ill-posed (e.g. non-unique solutions). Although solution non-uniqueness may be addressed by performing multiple tests on a structure, it is not trivial to decide which tests to carry out given that actual physical testing is costly. This problem is addressed in this work through a new co-evolutionary algorithm that interactively searches for damage scenarios and optimum physical tests. The algorithm is composed of two stages: the estimation phase, which searches for damage scenarios that can predict current physical tests, and the exploration phase, which searches for tests that increase the level of information about the damaged system. The feasibility of the methodology is demonstrated using numerical examples. Copyright © 2006 John Wiley & Sons, Ltd.

Received 27 June 2005; Revised 19 May 2006; Accepted 20 May 2006

KEY WORDS: genetic algorithms; damage identification; co-evolution; structural health monitoring; inverse problems

1. INTRODUCTION

Structural health monitoring systems use a combination of sensing, actuation, and reasoning components to infer the location and extent of damage in a structure. One of the main challenges in

*Correspondence to: W. Aquino, 313 Hollister, Civil Engineering Department, Cornell University, Ithaca, NY 14853, U.S.A.
†E-mail: wa27@cornell.edu
‡E-mail: bk84@cornell.edu
§E-mail: jb382@cornell.edu
¶E-mail: hod.lipson@cornell.edu

Contract/grant sponsor: U.S. National Science Foundation CAREER; contract/grant number: 0547376
Contract/grant sponsor: Cornell University

damage identification is to obtain sufficient information through sensing so that reasoning systems can uniquely and unambiguously characterize damage. One approach to address this challenge is to devise strategies for optimizing the location of sensors in order to maximize the information and increase the probability of correctly detecting damage [1, 2]. Another strategy is to actively search for damage through actuation and sensing. However, no logical method exists for deciding how to perform tests (i.e. actuation and sensing) that maximize the likelihood of correctly identifying damage.

In this work, a new methodology for structural damage detection is presented. In this new method, the damage detection process is carried out using co-evolutionary algorithms that are designed to maximize the information exploited from the system on each physical test. The predator–prey paradigm of co-evolution is used to set up competition between tests and damage hypotheses with the ultimate goal that false hypotheses are eliminated and the correct solution is found. To the best knowledge of the authors, a method for interactively guiding the testing process in damage identification applications is inexistent in the current literature.

2. BACKGROUND

The process of structural damage identification is commonly cast as an optimization problem in which an error measure that defines the discrepancy between computed and observed structural responses is minimized by updating a parameterized model [3]. Genetic algorithms have been successfully used for error minimization in various damage identification problems [2, 4–8] and have gained wide acceptance due to their inherent advantages such as convergence to global optima, noise tolerance, handling of multimodal problems (non-unique solutions), and gradient-free search.

The identification of damaged elements from measured structural response can be formulated mathematically as an inverse problem. As most inverse problems, damage identification may be ill-posed in the sense that more than one solution can exist (i.e. non-uniqueness) and/or the solution may not be continuously dependent on the input data (i.e. stability) [9]. In the case of structural damage identification problems, ill-posedness translates into many different possible damage scenarios and sensitivity of the solution to imperfections in the measured data (e.g. sensor noise, modelling errors, etc.).

Non-uniqueness of the solution in damage identification problems commonly arises from the sparsity of the measurements. In real applications, not all the degrees of freedom in a structure are usually measured. This gives rise to situations in which the number of parameters to be identified is larger than the number of measurements, resulting in a large number of possible solutions. Moreover, even if all degrees of freedom were measured, the static indeterminacy of skeletal structures (i.e. frames and trusses) would also lead to ill-posed inverse problems [5]. This situation is even more evident in damage identification problems that use finite element analysis of continua.

To address the non-uniqueness pathology, multiple tests are necessary to discard false hypotheses. Chou and Ghaboussi [5] and Hjelmstad and Shin [3] used multiple static tests to detect damage in truss structures. However, the approach for selection of tests in their work was not general (i.e. success depends on damage scenarios and type of structure). Other investigators, such as Limsamphancharoen [10], have recognized the need of multiple tests for reliably detecting damage in structures.

One of the major limitations so far in performing multiple tests for damage identification is the lack of *a priori* knowledge about damage. Random selection of tests may result in data sets that are

DOI: 10.1002/nme

not independent (e.g. contain redundant information) or do not contain enough information about the solution, resulting in false identifications. One of the key issues in damage identification is the exploitation of information contained in the sensed response. Here a co-evolutionary methodology for intelligently generating tests is proposed and demonstrated.

Co-evolutionary strategies have recently been proposed for non-linear system identification [11]. Co-evolution is a biological process where populations of interacting individuals challenge each other in an ongoing cycle of adaptation. Co-evolution can take many forms: the antagonistic arms-races of a predator and prey, the symbiotic co-operation between individuals in an ecosystem, or the mutual exploitation of a host and a parasite. There has been growing interest in co-evolutionary algorithms within the evolutionary computation community, starting with the seminal work of Hillis [12] on sorting networks. Furthermore, co-evolutionary algorithms have also been used in structural and architectural design [13, 14]. Contrary to conventional evolutionary systems, in which individuals are evaluated using a static quality or fitness metric, co-evolutionary systems consist of one or more populations in which individuals may influence the relative ranking of each other [15]. A new methodology based on co-evolution is presented herein for damage identification using minimal physical testing [11]. The proposed methodology, referred to as 'the estimation-exploration algorithm (EEA)' from hereon, provides an intelligent approach for selecting multiple tests in damage identification problems so that information contained in the measured response is maximized. In the co-evolution process, models evolve over generations to predict current tests, while current tests evolve to create discrepancy among probable damage hypotheses.

3. FORMULATION OF THE PROBLEM

The main problem consists in determining the location and extent of damage in a structure with a minimum number of physical tests. A test is defined as a set of loads applied to the structure and the location of the sensing points. It is assumed that a small number of sensors and actuators are used. In general, a small number of sensors will result in non-unique solutions unless multiple tests are performed, as discussed previously. Skeletal structures, specifically truss bridges, under static loading are considered in this work for the sake of simplicity and clarity. As described by Hjelmstad and Shin [3], the use of static response simplifies the theoretical framework for demonstrating the feasibility of damage identification algorithms. In addition, static tests eliminate uncertainties related to insufficient knowledge about the damping and mass distribution in the structure as is the case in vibration tests.

For the skeletal structures studied in this paper, the mathematical model for the system can described as

$$\mathbf{K}(\boldsymbol{\alpha})\mathbf{u} = \mathbf{f} \tag{1}$$

where $\mathbf{K}(\boldsymbol{\alpha})$ is the stiffness matrix of the structure defined as a function of a vector of damage parameters $\boldsymbol{\alpha}$, $\mathbf{f}$ represents the load applied to the structure, and $\mathbf{u}$ are the computed displacements at all degrees of freedom. There is one damage parameter, α, per element of the truss that represents the relative decrease in stiffness of that element. The reduced stiffness, $\mathbf{K}_i^{\mathbf{ee}}$, corresponding to element i in the structure can be computed from the undamaged stiffness, $\mathbf{Ko}_i^{\mathbf{ee}}$, and the corresponding damage parameter, α_i, as

$$\mathbf{K}_i^{\mathbf{ee}} = (1 - \alpha_i)\mathbf{Ko}_i^{\mathbf{ee}} \tag{2}$$

DOI: 10.1002/nme

The global stiffness matrix is assembled from individual element contributions as

$$\mathbf{K}(\boldsymbol{\alpha}) = \sum_{\text{elements}} \mathbf{K}_i^{\mathbf{ee}} \tag{3}$$

Damage parameters are obtained by minimizing the error between the computed response and the measured displacements. The error function used in this article is given by

$$E(\boldsymbol{\alpha}) = \frac{1}{n} \sum_{j=1}^{n} \left\| \frac{\mathbf{u}_{\mathbf{c}}^j - \hat{\mathbf{u}}^j}{\max(\hat{\mathbf{u}}^j)} \right\| \tag{4}$$

$$\mathbf{u}_{\mathbf{c}}^j \subset \mathbf{u}^j$$

where n is the number of tests performed on the structure, j indexes the tests, $\mathbf{u}_{\mathbf{c}}^j$ is a vector whose elements are the computed displacements at sensor locations, and $\hat{\mathbf{u}}^j$ is a vector containing the measured displacements. The difference between displacements should be normalized to get a better representation of the relative change in response. The maximum displacement of all sensors is selected as the normalization parameter to cancel out spurious effects of very small displacements. The damage parameters, $\boldsymbol{\alpha}$, are obtained by solving the following optimization problem:

$$\underset{\boldsymbol{\alpha} \in \mathbf{R}^m}{\text{Minimize}}\ E(\boldsymbol{\alpha}) \quad \text{Subject to } 0 \leqslant \alpha_i \leqslant 1 \tag{5}$$

4. THE ESTIMATION-EXPLORATION ALGORITHM IN STRUCTURAL DAMAGE IDENTIFICATION

The problem of damage identification is concerned with existence, location, and severity of the damage in a structure. When damage identification is cast as an optimization problem, great challenges arise such as large search domains and lack of sufficient information from tests performed on the structure. As explained before, this leads to ill-posed inverse problems in which solution uniqueness is not guaranteed. The naïve way of getting around this situation is to continue to test the structure until enough information can be extracted from the structure. However, apart from the cost, blind testing may fail to elucidate sufficient information about the damage state of the structure, even if a very large number of tests were carried out.

The EEA presented in this work is designed to search for the global optimum solution with the least number of physical tests. Figure 1 shows a flowchart that summarizes the steps that comprise EEA. Two main stages are involved in this algorithm: the estimation phase and the exploration phase. In the estimation phase, candidate solutions that minimize the error norm defined in Equation (4) are sought based on information gained from current tests. In the early stages of EEA, multiple solutions or candidate damage scenarios will be obtained due to the ill-posedness of the inverse problem. In the exploration phase, a set of the best candidate solutions found in the estimation phase are used to select the next test to be performed on the structure. The exploration phase is cast as an optimization problem in which the objective is to maximize the discrepancy among candidate damage scenarios. Maximizing the discrepancy among candidate damage scenarios can be interpreted as increasing information about the state of the structure since the selected candidate solutions can already explain all the existing test data. The goal of the exploration phase is to reduce the number of false hypotheses and direct the estimation phase towards the optimum solution.

DOI: 10.1002/nme

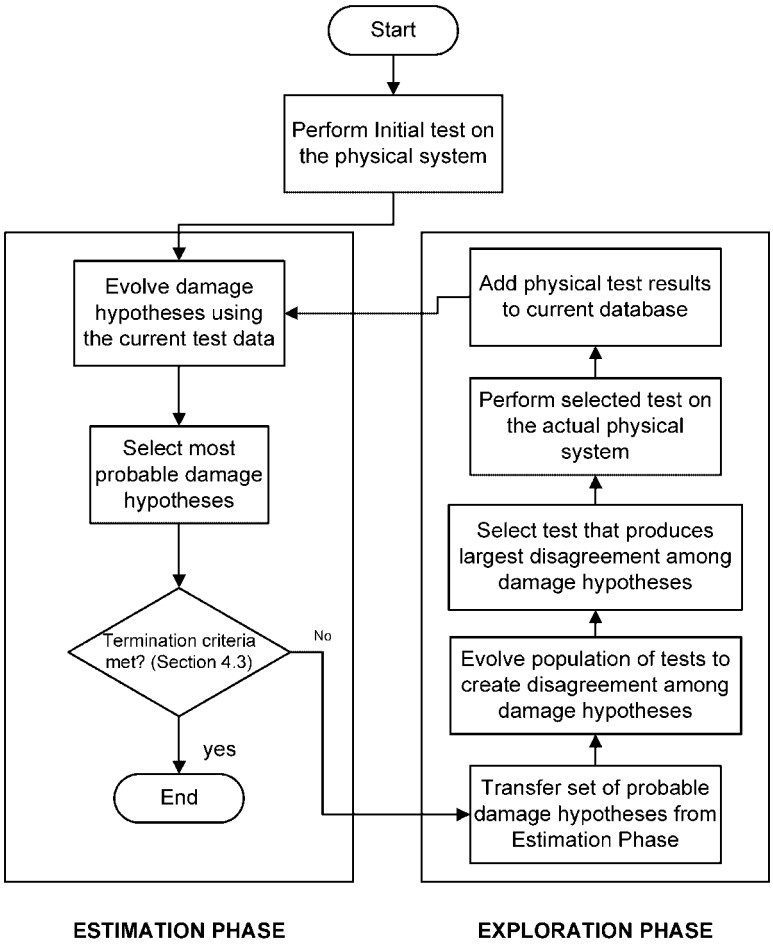

Figure 1. Flow chart for estimation-exploration algorithm.

Genetic algorithms are used in this work for both the estimation and exploration phase. A cycle is defined here as one estimation pass plus one exploration pass. In the case of lack of prior knowledge about the structure, the exploration phase starts with a randomly selected test that is performed on the actual physical system, while the estimation phase starts with a randomly initialized population of models. After the first estimation-exploration cycle is completed, the user has two options for the initial population of the estimation phase: (a) use the best individuals from the current population and randomly generate the rest of the population, (b) carry the entire population of models throughout the entire co-evolution process, and generate random individuals just for the first cycle. Approach (a) was used in this research. After the first estimation phase, the population at the end of a cycle was ranked by fitness (from best to worst) and the top 30% of the population was transferred to the next cycle. The other 70% of the population was then randomly generated. This is an elitism approach that guarantees that information learned from previous tests

DOI: 10.1002/nme

will be preserved, while the random generation of part of the population helps in exploring new areas of the search space and maintaining diversity.

4.1. Estimation phase

The estimation phase consists in finding candidate models that minimize the error between the predicted response and the tests that have been carried on the structure so far. In our work, the fitness of each individual is defined as the negative of the output error. The following fitness function is used in the estimation phase:

$$f = -E(\boldsymbol{\alpha}) \tag{6}$$

The population of solutions is evolved for a given number of generations and the best z candidate damage scenarios are transferred to the exploration phase (i.e. selection of tests). Here z is a number defined by the user and represents the number of candidate solutions that will be used for selecting the next test.

4.1.1. Solution encoding. The encoding of solutions is one of the most important parts of genetic algorithms (GA) and can have a great impact on the performance of this method. The simplest approach as related to damage identification is to encode damage indexes for all elements in the structure in a single chromosome (i.e. array of genes). However, it has been shown by Chou and Ghaboussi [5] that this is not a feasible approach. A more efficient method is to assume that the number of damaged elements in the structure is likely to be much smaller than the total number of elements and devise an encoding strategy accordingly. Chou and Ghaboussi used a solution representation based on binary strings in which only a small subset of the elements of the structure is assumed to be part of the solution. This approach, called the implicit redundant representation (IRR), produces much less variance in the solutions and has proven to be more reliable than searching for damage indexes in all elements at once [4].

As in IRR, the method used in this article uses prior knowledge about the probability of having a small number of damaged elements. However, real-number chromosomes are used as opposed to the binary representation used by IRR. Each solution in the current approach is composed of two chromosomes. The first one, termed the 'identifier chromosome' from here on, determines whether or not damage is present in an element, while the second one, called the 'damage chromosome' from here on, encodes the damage ratios for all elements.

Solutions are parsed by evaluating entries in the identifier chromosome first and then the ones in the damage chromosome. The present encoding method is illustrated in Figure 2. String 1, the identifier chromosome, determines if a specific member of the structure is damaged. It contains real numbers in the range [0, 1]. If the value of any gene in a chromosome is less than a predefined parameter m, the corresponding member is assumed to be damaged. Otherwise, a damage ratio of zero (i.e. undamaged case) is assigned to the corresponding gene in String 2. The parameter m is defined by the user and represents an approximation to the expected number of damaged elements. Notice that m does not define the number of damaged elements; it just assigns a probability of having a certain number of damaged elements.

String 2 contains the actual damage ratios, but only the genes for which damage has been identified in String 1 are active. It is important to realize that this method can still encode solutions

DOI: 10.1002/nme

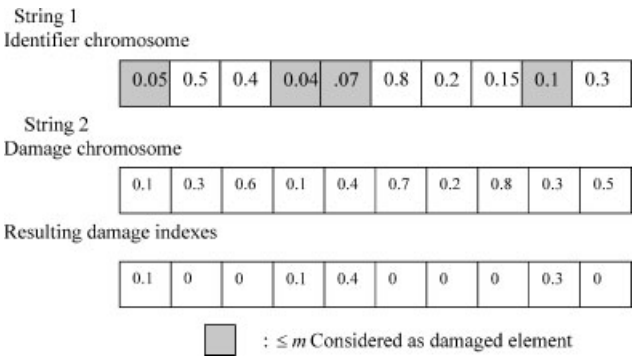

Figure 2. Solution encoding, using threshold $m = 0.1$.

ranging from no damaged element present to the extreme case of having all elements damaged. By using a damage probability parameter m, the dimension of the search space is regulated such that there is a higher probability for the solution to lie in the range of the expected number of damaged elements.

Notice that the solution encoding proposed in this paper has a noticeable advantage over a simple one-string encoding. If the value m drifts above its threshold for damage, the damage chromosome can still maintain the damage level, which could be reactivated later.

4.1.2. Genetic operators. Crossover and mutation were used as genetic operators. The deterministic crowding method (DCM) and a modified fitness sharing strategy devised by the authors were used for pairing individuals and producing offspring, while maintaining diversity of the population. A two-point crossover was used in the estimation phase. Like its counterpart in nature, mutation is responsible for introducing new solution states so that a larger extent of the search domain is explored. The mutation operator described in Reference [16] uses this property for fine tuning the solution. In this operator, uniformly distributed random numbers in the range $[-1, 1]$ are added to or subtracted from each gene with a probability defined by the user. In the event that the resulting number is greater than one, the value is capped at this upper limit, and if it is less than zero, it is assigned a value of zero.

4.1.3. Maintaining diversity. One of the key issues in EEA is to maintain diversity of the candidate damage scenarios throughout the entire process. Diversity is critical in EEA since selection of good tests relies on their ability to create disagreement among candidate damage scenarios. Diversity also prevents loss of potentially good information during the estimation phase and helps the exploration stage by producing candidate damage scenarios that carry different information about the state of the structure. There exist several methods that have been proposed for maintaining diversity in genetic algorithms. These methods are usually referred to as niching techniques and include crowding methods, neighbourhood methods, and fitness sharing among others [17].

The DCM [17] was used in this work. Since diversity is critical for the success of this algorithm, details of its implementation are provided here, though other diversity maintenance methods can also be used. In DCM, individuals are randomly selected (selection without replacement) and

DOI: 10.1002/nme

paired for the reproduction stage. Crossover and mutation are applied to the parents to get two offspring. Then, each child is compared to one of the parents and the individual with the higher fitness is transferred to the population of the next generation. The comparison is made between the child and the parent that are most similar (based on Euclidian distance of the chromosomes). This ensures keeping the best individuals found in the current generation (Elitism) and helps in maintaining diversity by giving less chance to similar individuals to survive.

The pairing of an offspring and a parent in DCM is decided as follows. Let D_{ij} be the Euclidian distance between Parent i and Child j. The following algorithm is used then for pairing the parents and the children:

```
if D11 + D22 < D12 + D21 then
    compare Parent 1 with Child 1
    compare Parent 2 with Child 2
else
    compare Parent 1 with Child 2
    compare Parent 2 with Child 1
end
```

It was found that, although deterministic crowding is able to maintain diversity for a number of generations, it sometimes succumbs to the intense drive of current strong individuals in the population, which are not necessarily good solutions in the early stages of co-evolution. For this reason, a modified version of fitness sharing was also used to strengthen diversity. The proposed fitness sharing forms groups of individuals based on a fixed Euclidian distance between damage chromosomes. The best individual in the group is kept without any change, but the fitness of the other members is divided by the number of individuals in that group.

4.2. Exploration phase

The goal of the exploration phase is to find a test that maximizes discrepancy among candidate solutions (i.e. damage scenarios) selected from the estimation phase. This task is cast as an optimization problem, which is solved using a genetic algorithm. The fitness of a test in the exploration phase is proportional to its ability to create disagreement among candidate solutions. The fitness function for tests used in this work is defined as

$$f_{\text{test}} = 100 \cdot \sum_{b=1}^{m\,\text{dof}} \sqrt{\frac{1}{z} \sum_{a=1}^{z} \left[\frac{u_{ab} - \overline{u}_b}{u_{\max}} \right]^2} \tag{7}$$

where z is the number of candidate solutions transferred from the estimation phase, u_{ab} is the displacement at degree of freedom b corresponding to model a, $\overline{u}_b$ is the average displacement obtained from the candidate models at degree of freedom b, $u_{\max}$ is the maximum displacement computed from all models, and m dof is the number of degrees of freedom where

DOI: 10.1002/nme

displacements are measured. Equation (7) is in essence the summation of the standard deviations of displacement at each degree of freedom for all selected models. It is important to keep in mind that selected candidate solutions can predict the data collected from the tests performed so far. Therefore, by creating discrepancy among candidate solutions, hidden information about the state of the damage structure is revealed.

The population of tests is randomly generated at the beginning of the exploration phase. Then, selection, crossover, and mutation are applied over a number of generations. At the end of the evolution process, the test genome with the highest fitness is selected and is implemented in a physical experiment. Results of this experiment are added to the existing bank of tests and the estimation phase is invoked for the next cycle of the algorithm.

4.2.1. Test encoding. Different strategies may be adopted for encoding tests during the exploration phase. These strategies depend on the type of excitation (e.g. static versus dynamic tests) and quantities being measured. For instance, in the case of static loads, a test may be defined by the number of forces applied to the structure, the direction, location and magnitude of the forces, the number of degrees of freedom being measured, and the location of the sensors. Whatever test definition is used, an important issue to always consider is the conservation of building blocks in the encoding scheme in order to maximize the effectiveness of the evolution process. For instance, the encoding should assure that blocks of sensors and forces are transferred between individuals during crossover. The reason for this is that sensor locations that can detect localized damage depend on the forces acting on the structure.

For simplicity, it is assumed in this paper that force magnitude is constant, and that the number of sensors and forces that define a test are fixed. Each test can be fully defined by the location and direction of the applied forces and the location of the sensors. Notice that fixing the magnitude of forces and the number of sensors and forces is not required by EEA. The methodology is general and different encodings can be devised for any conceivable structural test.

The test encoding used for the truss examples presented in this work can be described as follows. An individual test is defined by a chromosome composed of genes with integer values, which define position of forces and sensors. The forces can be applied in any of two directions, X or Y. Let N be the total number of nodes in the structure. Test genomes encode integers in the interval $[1, 3N]$. Three different ranges of integers are used to define the location of forces in the X- and Y-direction, and the location of sensors. Let P be the value of a gene in a test genome. This entry is interpreted according to the following convention:

if $P \leqslant N$

    Apply force in **X**-Direction at Node P

else if $N < P \leqslant 2N$

    Apply force in **Y**-Direction at Node $(P - N)$

else

    P defines a sensor location at Node$(P - 2N)$

end

DOI: 10.1002/nme

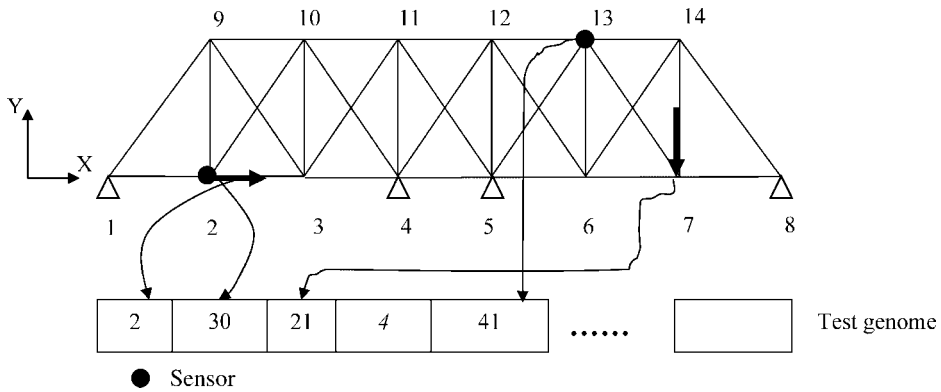

Figure 3. Test encoding.

The initial population of tests is generated randomly. Each test genome contains $3N$ genes. For each finite element run, a test is parsed by reading the locations of forces and sensors in the genome from left to right until the specified number of forces and sensors are obtained. The rest of the entries in the string are ignored. In addition, the algorithm checks for support conditions at nodes so that no force or sensor is specified at these locations. It is important to realize that for each test, a finite element analysis is performed for each of the candidate damage scenarios obtained in the estimation phase.

Consider the example shown in Figure 3 for which $N = 14$. Two forces and two sensors are specified in this figure. The test genome is traversed twice: one for locating forces and another for locating sensors. In this example, the first ($P = 2$) and third ($P = 21$) positions correspond to a force at Node 2 in the X-direction and a force at Node 7 in the Y-direction, respectively. Sensor locations are found in the second ($P = 30$) and fifth ($P = 41$) positions, corresponding to Node 2 and Node 13, respectively.

4.2.2. Genetic operators. As in the estimation phase, individuals are paired randomly, without replacement, to reproduce. The genetic operators used in the exploration phase were a simple two-point crossover and a simple mutation that swaps the location of genes in a chromosome based on a probability parameter defined by the user.

4.3. Stopping criteria

The proposed algorithm terminates when one of these conditions is met.

- One individual can explain all tests in a predefined number of consecutive cycles. This indicates that a potentially good solution has been found.
- Diversity is not maintained. At least two different individuals need to be transferred from the estimation phase to the exploration phase. Loss of diversity may be due to various factors such as inadequate parameters in the niching method, size of the population, number of generations, etc.
- The exploration phase cannot find a test that causes disagreement among candidate solutions. In this case, the algorithm cannot single out a unique solution. This situation may indicate that the problem is not fully observable.

DOI: 10.1002/nme

5. EXAMPLES USING NUMERICAL SIMULATION

The feasibility of the methodology is demonstrated with several numerical examples. A four-span bridge truss (Figure 4) with 105 elements and 44 nodes and subjected to static loading was used in all the examples. The area selected for the elements of the truss was 2500 mm^2 and the Young modulus of the material was 200 GPa. Five damaged elements localized in the first left bay were introduced in the structure. This element configuration was selected because it presents a more difficult damage identification problem than having elements scattered in the structure. Chou and Ghaboussi [5] demonstrated that damage identification in truss structures becomes more difficult when elements are connected at a node and only a limited number of degrees of freedom are measured. In addition, if tests are selected at random, it is less likely that sensors and loads will be placed in this localized area by chance. The level of damage induced in these elements is shown in Table I.

In addition to the proposed method (EEA), a control algorithm was used in which the exploration phase was disabled and random tests were used instead. Also, a case where tests were designed by the authors was investigated. In this case, 10 structural tests were engineered so that all bays were tested and loads and sensors were kept in close proximity to maximize the measured response. For this case, the damage identification process was carried using all the data collected from the tests at once as it is the common procedure in the literature. The control algorithm and the engineered tests serve as a benchmark to determine whether the proposed co-evolutionary strategy has significant advantages over these alternate approaches.

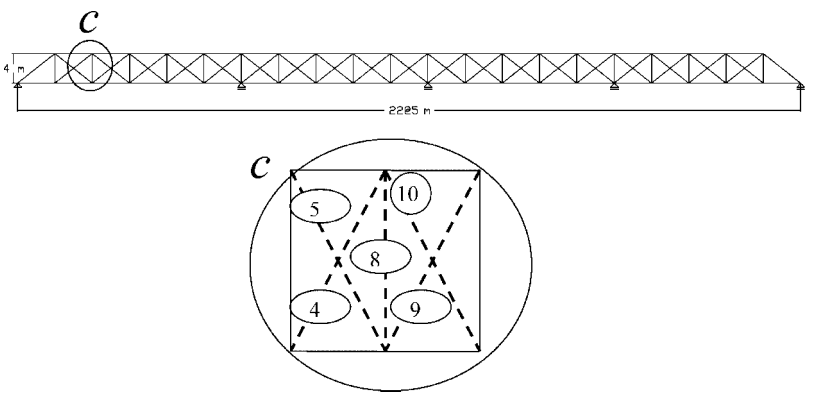

Figure 4. Truss structure used in numerical examples.

Table I. Damage ratios used in truss example.

Element label	Damage ratio, α
4	0.40
5	0.25
8	0.10
9	0.20
10	0.30

DOI: 10.1002/nme

Table II. GA parameters used in the examples.

Estimation phase	
Population size	200
Crossover probability (%)	100
Mutation probability (%)	10
Maximum number of tests (cycles):	10
Number of generations:	300
Exploration phase	
Population size	100
Number of generations	300

Each test encoded the locations of three loads applied in the vertical direction at nodes in the bottom cord of the truss along with the locations of a fixed number of sensors also in the bottom cord. For each test selection strategy (i.e. EEA, control, engineered), the identification process was carried out for two different cases: three and ten sensors with and without noise. The GA parameters used in this problem are given in Table II. These parameters were selected based on common heuristics found in the GA literature. Because of the stochastic nature of genetic algorithms, 10 runs were carried out for each example.

To study the sensitivity of EEA to noise, uniformly distributed random noise was added to measurements in the simulated tests as

$$\hat{u}^j = \hat{u}_o^j(1 + \beta e) \tag{8}$$

where $\hat{u}_o^j$ is the measured displacement at degree of freedom j without noise, β is the noise amplitude, and e is a uniformly distributed random variable in the range $[-1, 1]$.

The number of candidate damage scenarios (z) transferred to the exploration phase from the estimation phase in each cycle plays a fundamental role in EEA. It is important that at least two different damage scenarios be transferred to the exploration phase. In addition, it is important to realize that transferring a large number of damage scenarios to the exploration phase would increase the information content of the tests, but at the price of increased computational cost. In order to limit the computational cost, the authors found that transferring a maximum of five damage scenarios in each cycle produced good results. These five damage scenarios were selected by ranking the population of individuals from best to worst at the end of the estimation phase and then choosing up to five of the best candidates. It is imperative that the individuals selected represent different damage scenarios. Therefore, during selection the individuals were compared with each other using a Euclidian norm. The entire population was searched from best to worst until at least two different candidate damage scenarios were found.

5.1. Results and discussion

5.1.1. Simulated experiments without noise. The damage identification process was first carried out using three loads and three sensors without adding random noise. The performance of each strategy was evaluated by considering its accuracy in identifying the damage elements, accuracy in estimating the damage index for each element, and the number of misidentifications produced. A misidentification is defined as a non-damaged element for which the algorithm produces an average damage index greater than zero.

DOI: 10.1002/nme

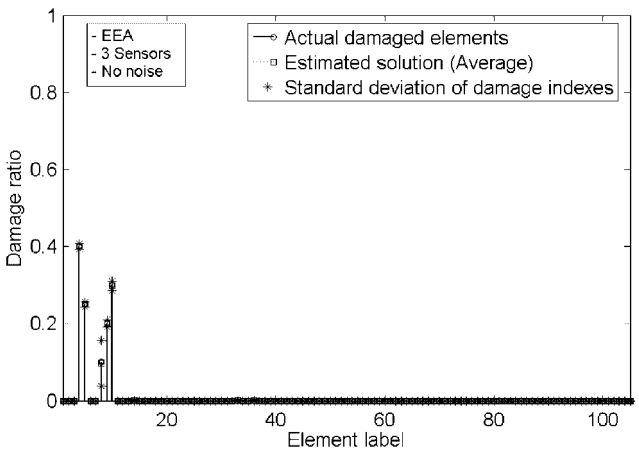

Figure 5. Damage indexes found using EEA after 10 cycles. No noise and three sensors.

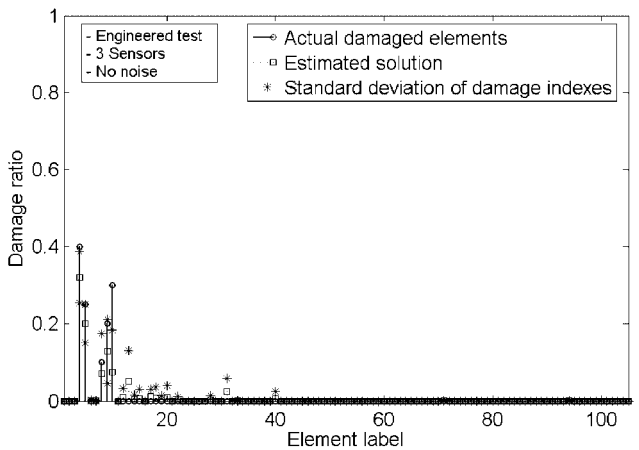

Figure 6. Damage indexes found using engineered tests. No noise and three sensors.

The results obtained from the different strategies (EEA, control, and engineered tests) are summarized in Figures 5–7. These plots show the average damage indexes found in each element as well as their standard deviation over ten computer runs. Figure 5 shows that EEA was able to find accurately and consistently all damage elements and their damage indexes at the end of 10 cycles. In addition, it produced no misidentification. Notice that only Element 8 presents some scatter in the results. It is interesting to notice that this is the only element that connects to all other damaged elements.

Figure 6 shows the results obtained when 10 engineered tests were used for the damage identification process. Figure 8 shows the sensor and load layouts of the engineered tests for the three-sensor case. It can be noticed in Figure 6 that the five damaged elements were correctly identified on average, but the EEA results were considerably more accurate in terms of the damage

DOI: 10.1002/nme

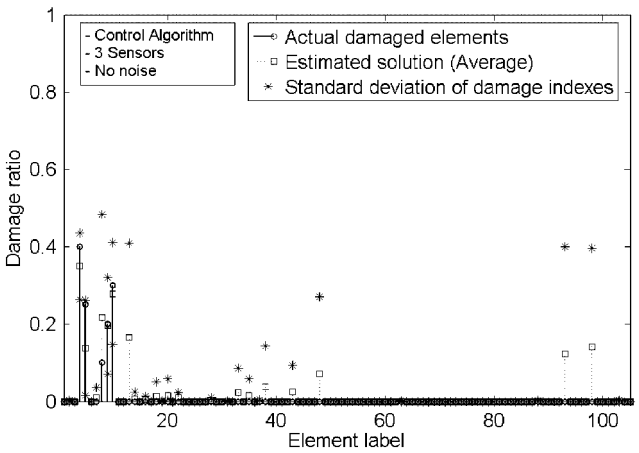

Figure 7. Damage indexes found using the control algorithm after 10 tests. No noise and three sensors.

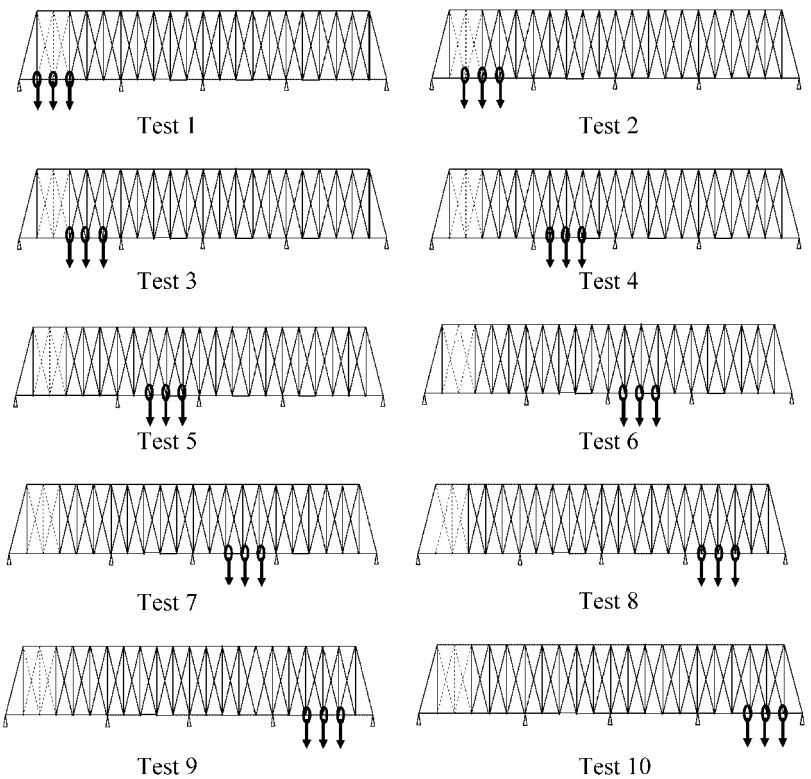

Figure 8. Sequence of engineered tests using three sensors per test.

DOI: 10.1002/nme

indexes. Notice also the larger scatter in the results obtained using engineered tests. There are several possible reasons for the superior performance of EEA over engineered tests. For instance, data in the engineered tests is presented all at once (as is common practice) to the damage identification algorithm, resulting in a more complex optimization problem. Also, EEA selects tests using a strategy devised for elucidating new information in each test, while engineered tests may contain redundant information.

Figure 7 shows the results obtained using the control algorithm in which location of sensors and forces in each test were generated randomly. It can be observed that although the five damaged elements could be found, the accuracy of the damage indexes is less than that observed for the solutions obtained by EEA and the engineered tests strategy. In addition, the results obtained by the control algorithm show a larger scatter and more misidentifications than those obtained with the other strategies. The poorer performance of the control algorithm is due to the random nature of the test selection. Many of the tests selected randomly may carry little information about the damage state.

In the discussion of co-evolutionary systems it is important to distinguish between the notion of objective error versus subjective error [18]. The subjective error refers to the ability of a solution (damage scenario) to predict the current data and can be defined as in Equation (4). The subjective error may be deceiving, especially in the early stages of the algorithm, as different damage scenarios may predict the observed experimental data equally well due to the ill-posedness of the problem. The objective error is a metric of how close the best individual is to the true solution. In this work, it is defined as the Euclidian distance between the damage chromosome of the best individual at the end of the estimation phase in each cycle and the true damage indexes. This can be expressed mathematically as

$$E_{\text{obj}} = \|\boldsymbol{\alpha}_{\text{sol}} - \boldsymbol{\alpha}_{\text{true}}\| \tag{9}$$

where $\boldsymbol{\alpha}_{\text{sol}}$ is a vector of damage indexes corresponding to the best individual of the population, E_{obj} is the objective error, and $\boldsymbol{\alpha}_{\text{true}}$ is a vector containing the true damage indexes for all elements. It is important to realize that this metric is not known in real life scenarios and was used here for the sole purpose of monitoring the progress of the EEA and its convergence behaviour towards the true solution.

Plots of the subjective error versus number of tests and objective error versus number of tests are shown in Figures 9(a) and (b), respectively, for both EEA and the control algorithm. Solid lines represent the average of the results taken over ten runs, while the error bars represent the average plus or minus one standard deviation. It can be observed in Figure 9(a) that for EEA the average subjective error increases with the number of tests and then decreases, while for the control algorithm the average subjective error increases continuously with the number of tests. EEA consistently reached a lower subjective error than the control algorithm by the end of 10 tests. In addition, EEA showed considerably less scatter in the results than the control algorithm. The objective error plots shown in Figure 9(b) illustrate how the best individual found by EEA approaches the true solution in fewer tests than the control algorithm, which indicates that the tests chosen by EEA were more informative than the tests randomly selected by the control algorithm.

5.1.2. Evolution of tests. An example of a sequence of tests and best damage scenarios found by EEA are shown in Figure 10. The stem plot in each figure corresponds to the damage scenario with the lowest subjective error found in the estimation phase. The location of sensors and loading in the first test was selected randomly. It can be observed in this example that after the first test

DOI: 10.1002/nme

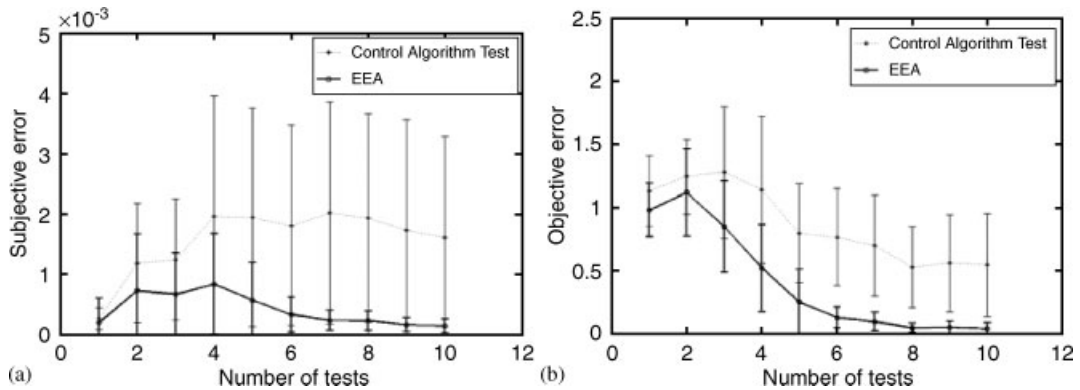

Figure 9. (a) Subjective error versus number of tests; and (b) objective error versus number of tests. Results are for EEA and control algorithm. Case for no noise and three sensors.

the best damage scenario significantly departs from the true solution despite being able to match the experimental data (low subjective error). This is a clear indication of the ill-posedness of the problem when very few measurements are available. In addition, due to this ill-posedness several damage scenarios with similarly low subjective errors were consistently found after the first few tests. The plots show that as more tests were performed on the system, false hypotheses were eliminated and the best damage scenario found by EEA approached the true solution. EEA was able to find the correct solution with only 6 tests in this example.

Examining the sequence of tests shown in Figure 10, it is noticed that in three tests (2, 3, and 6), EEA selected sensor locations away from the damage locations, while in the other two tests (4 and 5) some of the sensor locations were near the damaged area. It is expected from the structural viewpoint that tests performed near the damage area would reveal more information about the damage state than tests performed away from this area. However, it is important to realize that EEA uses disagreement among the current damage scenarios to select the next test. Therefore, some of the tests selected by EEA could place sensors and loads away from the damage area to eliminate false hypotheses. This behaviour was consistently observed throughout our numerical experiments.

5.1.3. The effect of noise. Random noise was added to the simulated data after each test during the identification process. The performance of EEA, the engineered tests strategy, and the control algorithm in the presence of noisy data using three sensors per test are depicted in Figures 11–13. It can be seen from these plots that noise decreases the accuracy of the algorithms and increases the scatter in the results as expected. Although the three strategies (EEA, control, and engineered tests) were able to locate the five damage elements, EEA was the most accurate in estimating the damage indexes and producing the fewest misidentifications. The control algorithm, as expected, produced the worst results, which can be attributed to the random test selection process as explained before.

In order to test the effect of using more sensors on the performance of the algorithms, the identification process was repeated using 10 sensors per test. The results for the three identification strategies (EEA, control, engineered) are summarized in Figures 14–16. Figure 17 shows the

DOI: 10.1002/nme

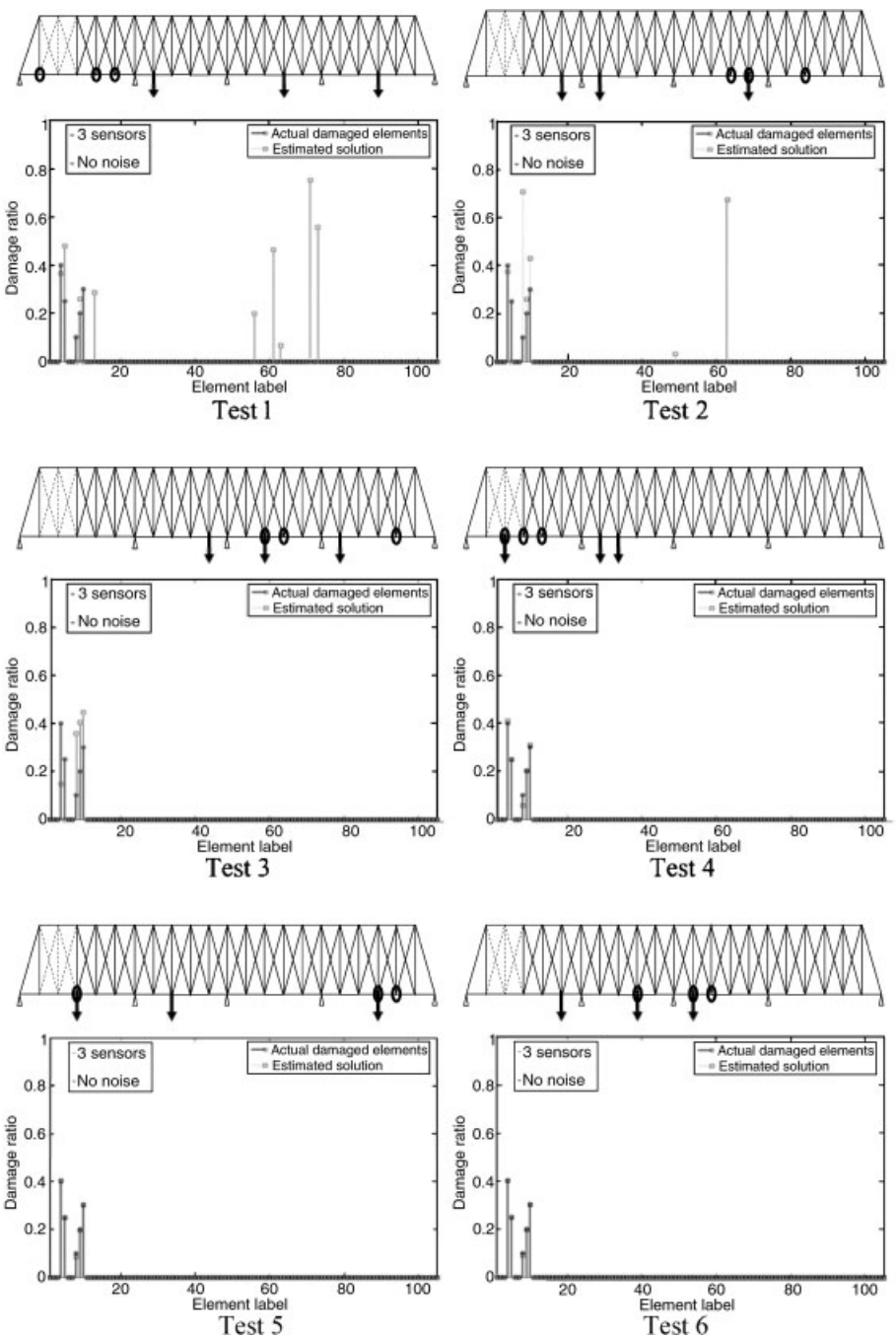

Figure 10. Example of a sequence of tests found by EEA. No noise three-sensor case.

DOI: 10.1002/nme

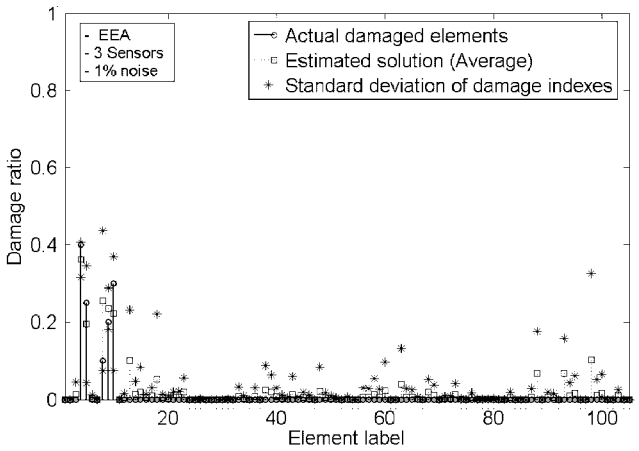

Figure 11. Damage indexes found using EEA after 10 cycles. Case for 1% noise and three sensors.

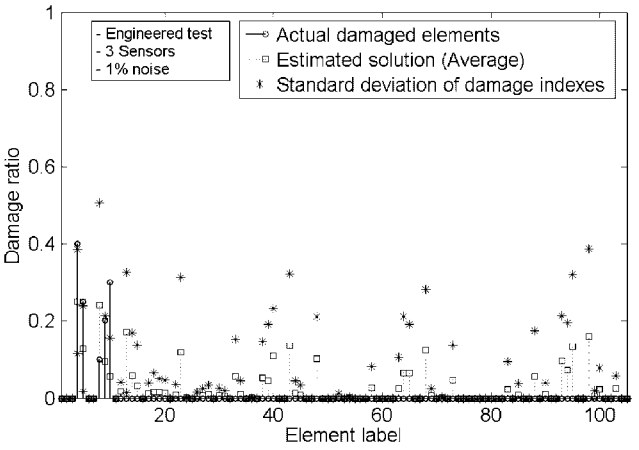

Figure 12. Damage indexes found using engineered tests. Case for 1% noise and three sensors.

sequence of engineered tests used to obtain the results shown in Figure 16. It can be observed in these figures that when 10 sensors where used, the three identification strategies produced more accurate results with less scatter and fewer misidentifications than when three sensors were used. This result was expected as the use of more sensors increases the level of information obtained in each test. The results also show that EEA produced more accurate results than both the engineered tests strategy and the control algorithm at the end of 10 tests. This can be inferred by noticing that the average damage indexes found by EEA are closer to the true solution than those found by the other two strategies. In addition, EEA produced fewer misidentifications than the engineered tests and the control algorithm did. Furthermore, notice that there was less scatter in the solutions found by EEA as compared to the solutions found by the other two strategies. As was observed previously, the control algorithm produced the worst results.

DOI: 10.1002/nme

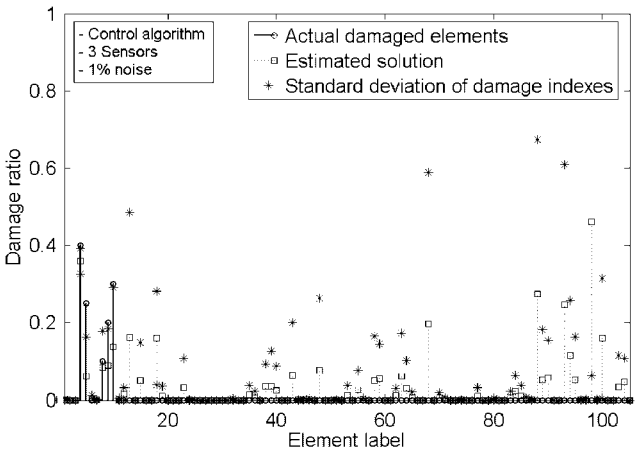

Figure 13. Damage indexes found using the control algorithm after 10 tests. Case for 1% noise and three sensors.

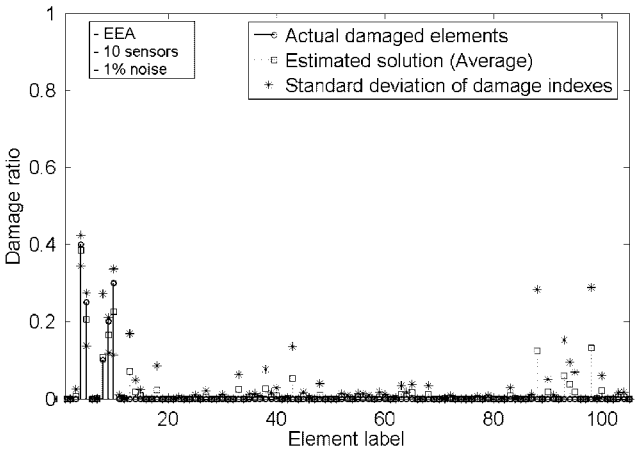

Figure 14. Damage indexes found using EEA after 10 cycles. Case for 1% noise and 10 sensors.

The subjective error plot and the objective error plot for the case of 10 sensors with noise are shown in Figures 18(a) and (b), respectively, for EEA and the control algorithm using noisy data. As opposed to the noiseless data case, it is interesting to observe that EEA consistently produced higher subjective errors than the control algorithm did. Notice that when noise is introduced in the data, even the correct solution would result in a non-zero subjective error. The lower subjective error displayed by the solution found by the control algorithm may indicate overfitting of the noisy data. The superior performance of EEA over the control algorithm is further demonstrated by the objective error behaviour, which was similar to the one observed in the noiseless data case. Figure 14(b) shows that the best damage scenario found by EEA approached the correct solution with fewer tests than the solution found by the control algorithm.

DOI: 10.1002/nme

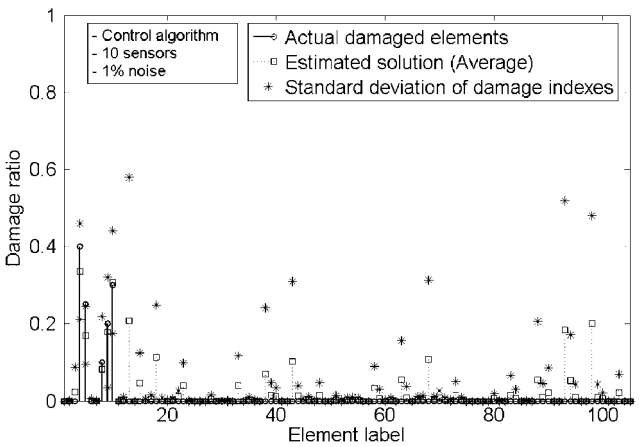

Figure 15. Damage indexes found using the control algorithm after 10 tests. Case for 1% noise and 10 sensors.

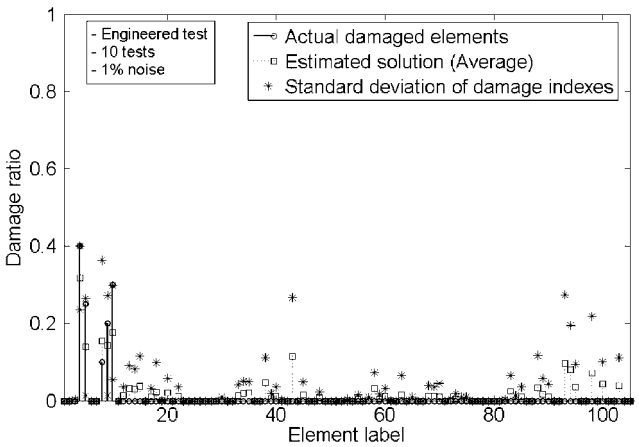

Figure 16. Damage indexes found using engineered tests. Case for 1% noise and 10 sensors.

5.1.4. Additional comments. The results shown above demonstrate the feasibility of EEA in its application to damage identification problems. The tests defined in the exploration phase of the algorithm were limited to a fixed number of sensors and loads. The proposed algorithm (EEA) is general and could also be used to evolve a variable number of sensors and loads. However, it is important to realize that this approach would result in a more complex and costly optimization problem since the search space for tests would increase. Despite increased computational costs, the evolution of a variable number of sensors and loads would be a highly desirable capability in real applications. The authors plan to incorporate this capability in future versions of EEA.

DOI: 10.1002/nme

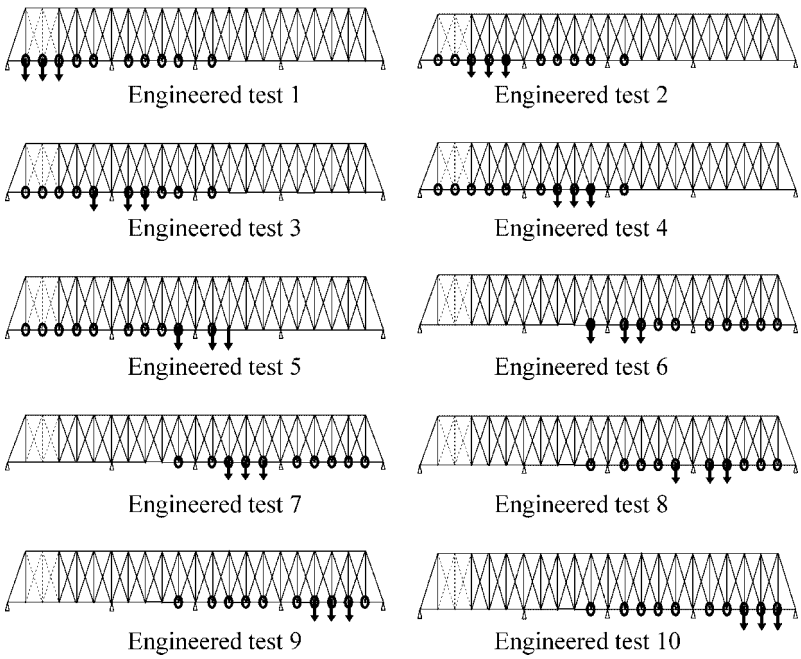

Figure 17. Sequence of engineered tests using 10 sensors per test.

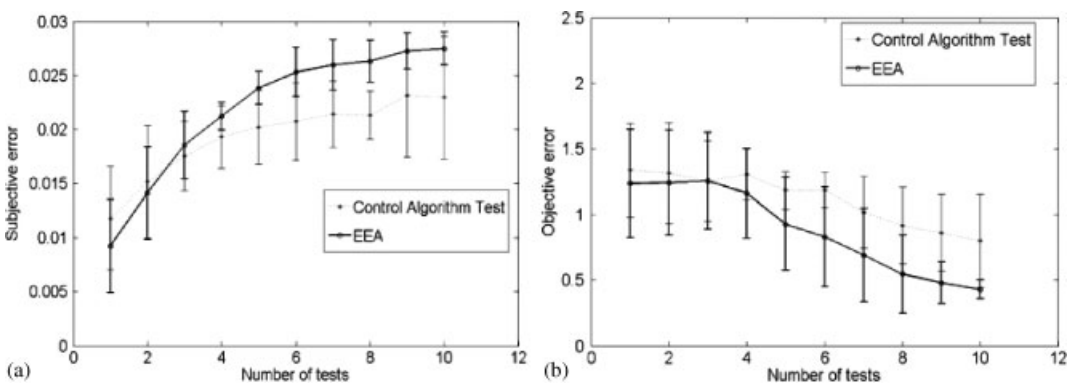

Figure 18. (a) Subjective error versus number of tests; and (b) objective error versus number of tests. Results are for EEA and random test selection strategy. Case for 1% noise and 10 sensors.

The results presented in this paper suggest that it would be more advantageous to present data to damage identification algorithms gradually, as it is done in EEA, as opposed to collect a copious amount of data and then present it at once to the damage identification algorithm.

Although EEA is still in its research and development stages, it has shown promise as potential tool for field investigations in the future. This implementation will require that an appropriate set-up for coupled computation and field-testing be devised. Since genetic algorithms are computationally

DOI: 10.1002/nme

expensive, one of the goals of the authors is to devise new surrogate models and parallel computing strategies that will permit the implementation of EEA in realistic field applications.

6. CONCLUDING REMARKS

In this article, a new co-evolutionary algorithm for structural damage identification has been presented. The algorithm is composed of two stages: the estimation phase that searches for damage scenarios and the exploration phase, which searches for tests that increase the level of information about the damage system.

The feasibility of the methodology was demonstrated through several numerical examples. The proposed algorithm was compared to two alternate strategies: a control algorithm in which the exploration phase was disabled and tests were generated randomly, and a strategy in which tests were engineered by the authors. EEA outperformed the control algorithm and the engineered tests strategy by displaying higher accuracy in identifying damage indexes and producing fewer misidentifications. In addition, the results obtained with EEA contained less scatter than those obtained with the two alternate methodologies. It was shown that EEA approached the true solution with fewer tests than the control algorithm and the engineered tests.

The proposed methodology is general and could be extended to other problems in damage identification and non-destructive testing. This approach would be particularly valuable in systems where little information exists about the damage state, few measurements can be obtained in each test, tests are very costly, and/or the space for test selection is very large. A wide variety of problems exists in real applications where one or more of these conditions are present.

ACKNOWLEDGEMENTS

Work by J. Bongard and H. Lipson was supported in part by U.S. National Science Foundation CAREER grant No. 0547376. Also, this research was funded in part by Cornell University.

REFERENCES

1. Papadimitriou C. Pareto optimal sensor locations for structural identification. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**(12–16):1655–1673.
2. Raich A, Liszkai T. Multi-objective genetic algorithms for sensor layout optimization in structural damage detection. *Proceedings of the Artificial Neural Networks in Engineering Conference*, 2003.
3. Hjelmstad KD, Shin S. Damage detection and assessment of structures from static response. *Journal of Engineering Mechanics* (ASCE) 1997; **123**(6):568–576.
4. Raich A, Liszkai T. Benefits of implicit redundant genetic algorithms for structural damage detection in noisy environments. *Proceedings of Genetic and Evolutionary Computation—GECCO*, 2003.
5. Chou JH, Ghaboussi J. Genetic algorithm in structural damage detection. *Computers and Structures* 2001; **79**(14):1335–1353.
6. Guo HY, Zhang L, Zhang LL, Zhou JX. Optimal placement of sensors for structural health monitoring using improved genetic algorithms. *Smart Materials and Structures* 2004; **13**(3):528–534.
7. Papadimitriou C. Optimal sensor placement methodology for parametric identification of structural systems. *Journal of Sound and Vibration* 2004; **278**(4–5):923–947.
8. Rao MA, Srinivas J, Murthy BSN. Damage detection in vibrating bodies using genetic algorithms. *Computers and Structures* 2004; **82**(11–12):963–968.
9. Liu GR, Han X. *Computational Inverse Techniques in Nondestructive Evaluation*. CRC: Boca Raton, FL, 2003.

DOI: 10.1002/nme

10. Limsamphancharoen N. Condition monitoring of structures by using ambient dynamic responses. Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 2003.
11. Bongard JC, Lipson H. Nonlinear system identification using co-evolution of models and tests. *IEEE Transactions on Evolutionary Computation* 2005; **9**(4):361–384.
12. Hillis WD. *Co-evolving Parasites Improve Simulated Evolution as an Optimization Procedure*. *Artificial Life II*. Addison-Wesley: Reading, MA, 1992.
13. Poon J, Maher ML. Co-evolution and emergence in design. *Artificial Intelligence in Engineering* 1997; **11**(3):319.
14. Maher ML. A model of co-evolutionary design. *Engineering with Computers* 2000; **16**(3–4):195.
15. Bucci A, Pollack JB, De Jong ED. Automated extraction of problem structure. *Proceedings of the Genetic and Evolutionary Computation Conference*, Springer-Verlag: New York, 2004.
16. Chipperfield A, Fleming P, Pohlheim H, Fonseca C. *Genetic Algorithm Toolbox*, *User's Guide*.
17. Mahfoud SW. Niching methods for genetic algorithms. University of Illinois at Urbana-Champaign, Urbana, IL, 1996.
18. Watson RA, Pollack JB. Co-evolutionary dynamics in a minimal substrate. *Proceedings of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann, San Francisco, CA, 2001.

DOI: 10.1002/nme