ASIAN JOURNAL OF CIVIL ENGINEERING (BUILDING AND HOUSING) VOL. 11, NO. 6 (2010)
PAGES 717-740

AN IMPROVED GENETIC ALGORITHM USING SENSITIVITY ANALYSIS AND MICRO SEARCH FOR DAMAGE DETECTION

S.S. Naseralavi, J. Salajegheh, E. Salajegheh* and M.J. Fadaee
Department of Civil Engineering, University of Kerman, Kerman, Iran

ABSTRACT

We present a new strategy for applying to continuous genetic algorithm for damage detection of structures. This strategy pursues two aims: 1) reducing search space by elimination of some design variables during optimization process, 2) improving each individual by solving the linearized problem using Moore-Penrose pseudo inverse at the end of reproduction of genetic algorithm. To these ends, two sub-programs are embedded after typical GA operators. This strategy is applied to three different types of problems: damage detection by frequencies and by static measurements, and crack identification of a beam using frequencies. Numerical results demonstrate the high efficiency of the proposed algorithm compared to those found in the literature.

Keywords: Structural damage detection; sensitivity analysis; continuous genetic algorithm

1. INTRODUCTION

Damage detection has attracted a lot of interest in recent years. The necessity of industry to health monitoring and curing makes damage detection techniques one of the most active fields of research in the recent years. These techniques, in which the damages are identified through a non-destructive test (NDT) instead of visual or local experimental techniques, have been successfully applied to many practical problems. Although damage is a nonlinear phenomenon, however, it is widely accepted that the damage can be simulated by changing in some parameters such as Young's modulus, cross-sectional area, moment of inertia and/or boundary conditions, etc. In this work the general damage is modeled by reduction in the Young modulus, while the edge crack as a special damage is simulated using a hinge and rotational spring.

In fact damage detection of structures is solving a system of nonlinear equations. Conventionally, such problems can be solved using iterative methods such as Newton-Raphson, arc-length method … or the optimization methods. Practically, in damage detection problems the number of equations (measured parameters) is usually less than the unknown variables (elemental damaging). Hence, this is an undetermined problem in

* E-mail address of the corresponding author: eysasala@mail.uk.ac.ir (E. Salajegheh)

mathematics and has infinite solutions. However, there are two useful additional conditions for seeking the right damage solution: 1) low l_2 norm condition: the right solution has low Euclidean norm (See Section 3), 2) low l_0 norm condition: in the damaged structure most of the structural elements are still intact, so the right solution has high sparsity.

For such problems, if the system of equations is linear, many mathematical solutions are available [1]. In the optimization based methods, damaged elements and damage extends are searched through an optimization process until the response of a hypothesized damaged structure equals to those of a real damaged structure. In recent years, genetic algorithm is frequently employed for damage detection and system identification [2]. In the case of large structures the optimization-based methods, which use the damaging of all elements as design variables, are computationally expensive. Therefore, many efforts have been devoted to reduce the size of search space in the optimization-based damage detection problems [2-6]. Wang et al. [6] localized the structural damages by matching frequencies and static responses of Measured Damage Signatures (MDS) with those of Predicted Damage Signature (PDS). And in the second stage a quadratic programming is sequentially iterated until the damages detected. Au et al. [3] obtained energy quotient difference by expanding the incomplete structural mode shapes, to find the most potentially damaged elements. Then, they detected the damages of the limited elements by micro genetic algorithm. For large scale structures, they also proposed a two level optimization method in which the subset of the damaged elements was searched. Guo and Li [4] restricted the design variables by applying evidence theory to frequencies and mode shapes data. Then, micro search genetic algorithm was employed for damage detection in which the vicinity points of the elitist of the generation is also searched. He and Hwang [5] first reduced the design variable of damage detection problem by grey relation analysis and furthermore, introduced an improved real coded genetic algorithm with a new mutation operator which merges the merit of simulated annealing. In addition, both mutation and crossover probability are increased for escaping from the local minimums. Ricardo et al. [7] first reduced the design variables by employing damage functions. After that, Strength Pareto Genetic Algorithm (SPGA) is used to optimize two selected objective functions. For damage detection with harmonic excitation response, Kokot and Zembaty [8] applied genetic algorithm to somewhat approach the structural damage situation. In their second stage Levenberg-Marquardt local search is applied to detect the damages. Sahoo and Maity [9] detected the damages by applying a neural network in order to perform reverse mapping to the faults of the damaged sub-structure. Genetic algorithm is utilized for optimal designing of their network architecture. A correlation-based damage detection approach has been presented by Messina et al. [10] in which multiple damage location assurance criterion (MDLAC) has been introduced to detect the multiple damages. The results were shown a good prediction for damage locations but with a small error for damage sizes. Further, Koh and Dyke [11] have employed genetic algorithm to maximize the MDLAC index for structural damage detection. A comparison between the sensitivity-based methods with the optimization-based methods has been made by Gomes and Silva [12]. In that study, a modified MDLAC has been used as the objective function for their optimization-based method. Both the methods were good to detect the damage sites, but not so good to predict the damage extents.

Lee [13] proposed a method for multiple crack detection in a beam based on Newton-

Raphson method in which treat the crack location and size as continuous design variables. The number of required natural frequency is declared to be double of number of cracks, and the method has been examined until triple cracks. The sensitivity matrix has been created using the finite difference method to be used in Newton-Raphson procedure. An application of the micro-genetic algorithm to detect a single crack in a real damaged beam has also been employed by Vakil-Baghmisheh et al. [14]. In another work, Caddemi and Morassi [15] presented analytical method for single crack detection by static measurements, and also obtained the required conditions for unique detection of the crack. Khaji et al. [16] proposed a closed form solution for single crack detection in Timoshenko beams which was tested by various boundary conditions.

In the present paper, the process of restricting the design variables is done inside the genetic algorithm body. In opposite to the works in the literature, at this study the elemental damage and crack detection are dealt simultaneously.

In the present work two additional operators are embedded inside the genetic algorithm which are to operate after usual genetic operators. In the first operator, Sensitivity Based Improvement (SBI), for each individual the sensitivity matrix of structural responses with respect to elemental damaging is first established to be solved by Moore-Penrose pseudo inverse. The individuals are updated by adding their corresponding obtained solutions to themselves. Now the updated individuals are ready to go to the second operator, named as Micro Search (MS). This operator works just after each several generations. The duty of MS is to limit the design variables by eliminating of the low variables of the elites. Therefore, by using these new operators, in addition to the structural response approaching to the damaged one, low l_0 and l_2 norm conditions are automatically satisfied.

This paper is organized as follows: The application of the sensitivity analysis for the damage detection is discussed in Section 2. Sensitivity of static displacements and frequencies of structures with respect to damaging is derived in Section 3. In Section 4 Continuous genetic algorithm is described. The proposed algorithm is presented in Section 5 and then three illustrative case studies are verified in Section 6. Finally, the conclusions end the paper in Section 7.

2. DAMAGE DETECTION USING SENSITIVITY-BASED ANALYSIS

To detect damages, the structural natural frequencies, mode shapes and static deflections or a combination of them may be used [6]. Due to damages, the stiffness of structure is reduced, while its mass is still remained constant. Therefore, its deflections due to external loading are increased, and also its natural frequencies are reduced and its mode shapes are changed. If the structure become damaged differently, the variations of deflections, natural frequencies and mode shapes will occurre in a different way. So, to detect the damages, one has to analytically find a damage state results in the same response as real damaged structure.

Thus, the problem of structural damage detection is formally equivalent to finding a set of damage variables by which the analytical responses of the structure match the measured ones in an optimal way. The mathematical expression of the problem can be defined as:

$$\mathbf{R}_d = \mathbf{R}(X) \;\Rightarrow\; \mathbf{X} = ?, \tag{1}$$

where $\mathbf{X} = \{x_1\ ,\ x_2\ , \ldots x_n\}^{\mathrm{T}}$ is called the damage vector, in which $0 \le x_i \le 1$ is damage ratio of the *i*th element, and *n* is the number of structural elements. The values $x_i = 0$ and $x_i = 1$ indicate the intact and completely damaged state, respectively. $\mathbf{R}_d = \{r_{d1}\ ,\ r_{d2}\ , \ldots r_{dm}\}^{\mathrm{T}}$ is the vector of *m* structural responses (the responses can be natural frequencies, complete and incomplete mode shapes, structural deformations due to static loadings or a combination of them) of the existing damaged structure and $\mathbf{R}(\mathbf{X}) = \{r_1(\mathbf{X})\ ,\ r_2(\mathbf{X})\ , \ldots r_m(\mathbf{X})\}^{\mathrm{T}}$ is the vector of *m* responses of a hypothetically damaged structure that can be evaluated from the analytical model.

Using the first order approximation Eq. (1) can be expressed as follows:

$$\mathbf{R}_d = \mathbf{R}(\mathbf{X}) = \mathbf{R}_h + \frac{\partial \mathbf{R}}{\partial \mathbf{X}} \Delta\mathbf{X} + \ldots \Rightarrow \mathbf{R}_d - \mathbf{R}_h = \Delta\mathbf{R} \simeq \mathbf{S}\ \Delta\mathbf{X}, \tag{2}$$

where $\mathbf{R}_h$ is the structural response vector of the healthy structure, $\Delta\mathbf{X}$ is damage vector change and $\mathbf{S} = \partial\mathbf{R}/\partial\,\mathbf{X}$ is the sensitivity matrix. Practically, the number of measured responses *m* is usually less than the unknown variables *n*. Hence, this is an undetermined problem in mathematics and its solutions are a subset, $S^* = \{\mathbf{z} : \mathbf{S}.\mathbf{z} = \Delta\mathbf{R}\}$. While **S** is a full rank matrix, we will have $dim(S^*) = n - m$. In usual damage detection problems, just a few numbers of structural elements are damaged, so $\|\mathbf{X}\|_0 << n$, where $\|\cdot\|_0$ denotes the number of nonzero entries. So, we seek for the solutions with high sparsity i.e. $\{\mathbf{X} : \|\mathbf{X}\|_0 \le k, \mathbf{R}_d = \mathbf{R}(\mathbf{X})\}$, where *k* is a value more than expected damages.

In fact $\mathbf{R}(\mathbf{X})$ is a continuous function of **X,** i.e. $\lim_{\mathbf{X}\to\mathbf{X}_d} \mathbf{R}(\mathbf{X}) = \mathbf{R}_d$. This continuity implies the correct solution has small distance from the origin. So, among all infinite solutions, the solution with minimum Euclidean norm (i.e. $\Delta\mathbf{X} \in S^*$ & $\Delta\mathbf{X} \perp S^*$) is taken as a primitive approximation for damage solution.

However, when $n < m$, the solution **X** will not lie exactly in $Span(\mathbf{S}) = \{\mathbf{S}.\mathbf{z} : \mathbf{z} \in R^n\}$, therefore an unbalanced vector $\mathbf{r} = \mathbf{S}.\Delta\mathbf{X} - \Delta\mathbf{R}$ always exist. However, while this unbalanced vector is perpendicular to the $Span(\mathbf{S})$, its magnitude will be minimized, and $\Delta\mathbf{X}$ can be determined as follows:

$$\begin{aligned} &\mathbf{r} = \mathbf{S}.\Delta\mathbf{X} - \Delta\mathbf{R} \perp Span(\mathbf{S}) \Rightarrow \mathbf{S}^T.\mathbf{r} = \mathbf{S}^T(\mathbf{S}.\Delta\mathbf{X} - \Delta\mathbf{R}) = 0 \\ &\Rightarrow \mathbf{S}^T\mathbf{S}\Delta\mathbf{X} = \mathbf{S}^T\Delta\mathbf{R} \\ &\therefore \Delta\mathbf{X} = \left(\mathbf{S}^T\mathbf{S}\right)^{-1}\mathbf{S}^T\Delta\mathbf{R}\ or\ \Delta\mathbf{X} = \mathbf{S}^+\Delta\mathbf{R}. \end{aligned} \tag{3}$$

In summary, regardless which of the values *n* or *m* is greater, the value $\Delta\mathbf{X} \simeq \mathbf{S}^{+}\ \Delta\mathbf{R}$ can be an improvement movement for the current point X toward the damage solution. The pseudo-inverse of $\mathbf{S}$ (i.e. $\mathbf{S}^{+}$) can be found by singular value decomposition (See Appendix I).

3. SENSITIVITY OF STRUCTURAL RESPONSES

3.1 Sensitivity of eigenvalues with respect to damaging

An approach for calculating the derivatives of eigenvalues with respect to damage ratio (or any other arbitrary design variable) of an element using the stiffness and mass matrices of the structure and the eigenvectors have been presented in Ref. [17], where:

$$\frac{\partial \lambda^{(i)}}{\partial x_j} = \boldsymbol{\phi}_i^T \frac{\partial \mathbf{K}}{\partial x_j} \boldsymbol{\phi}_i - \lambda^{(i)} \boldsymbol{\phi}_i^T \frac{\partial \mathbf{M}}{\partial x_j} \boldsymbol{\phi}_i, \tag{4}$$

in which $\boldsymbol{\phi}_i$ and λ_i denote the *i*th eigenvector and eigenvalue, and also $\mathbf{K}$ and $\mathbf{M}$ are the stiffness and mass matrices of the structure, respectively. In this study, since damaging is considered as reduction in elasticity modules of the elements, it will not affect the mass matrix, thus, the second term of the right side of Eq. (4) will be vanished.

3.2 Sensitivity of displacements with respect to damaging

The static equilibrium equation of the structure can be expressed as follows:

$$\mathbf{Kd} = \mathbf{F}, \tag{5}$$

where $\mathbf{K}$ is the stiffness matrix, $\mathbf{d}$ is the displacements vector and $\mathbf{F}$ is the applied loads vector. By differentiating Eq. (5) with respect to an elemental damage ratio (or any other design variable), the *i*th column of deformation's sensitivity matrix is obtained [6]:

$$\mathbf{K}'\mathbf{d} + \mathbf{K}\mathbf{d}' = \mathbf{0} \Rightarrow \mathbf{d}' = -\mathbf{K}^{-1}\mathbf{K}'\mathbf{d}\,.$$
$$\therefore \frac{\partial \mathbf{d}}{\partial x_i} = -\mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial x_i} \mathbf{d}\,. \tag{6}$$

It should be noted that for calculation of different columns of sensitivity matrix of displacements using Eq. (6), just ${\partial \mathbf{K}}/{\partial x_i}$ is changed and calculation of $\mathbf{K}^{-1}$, which devote the most computational effort, is executed once. Therefore, the computational effort of this way is so much lower than the conventional finite difference method.

4. CONTINUOUS GENETIC ALGORITHM (CGA)

Charles Darwin first inspired the process of natural evaluation and adaptation to environmental variation. Further, this idea simulated numerically by Holland as an optimization tool referred to as genetic algorithm. Since then, GA has been realized as an efficient method for stochastic global search and has been used in various aspects of civil engineering [18-19].

In the conventional genetic algorithm (discrete version) each solution point is coded as a binary string. But CGA also known as real coded genetic algorithm (RGA), directly uses the variables themselves which is so better for optimization problems with continuous variables. Coding and decoding process is not needed in CGA which saves much computational time. Generally in GA, each individual of the population is called chromosome and each variable is called gene. CGA is consisting of five main steps as follows:

1. Initialization: In this part, a set of primitive solutions as initial population is randomly created in the feasible region of the search space. For damage detection problems, the first generation can be obtained by generating a set of random damage vectors.
2. Fitness evaluation: In this part, the objective function for the individuals of current population is calculated. Here, for damage detection $\|\mathbf{R}(\mathbf{X})-\mathbf{R}_d\|$ is selected as the objective functions, where $\|\cdot\|$ indicates the Euclidean norm.
3. Selection: Herein, a set of individuals of the current population are copied and stored in a set referred to as math pool for reproduction process. The individuals possessing better fitness will have more copy in the math pool. The mat pool is also likening to routing wheel.
4. Reproduction: The reproduction process simulates the biological creation of a new generation. Reproduction process is usually consisting of two main stages: 1) Cross over and 2) Mutation.

Cross over is simulating marriage and generation of offspring by combining two individuals (chromosomes). Assume that $Chrom1=(x_1,x_2,\ldots,x_n)$ and $Chrom2=(y_1,y_2,\ldots,y_n)$ are two individuals from the current population, then one can gain the offspring, $Chrom1'=(x'_1,x'_2,\ldots,x'_n)$ and $Chrom2'=(y'_1,y'_2,\ldots,y'_n)$, by cross over operation from Eq. (7) :

$$x'_i=\alpha_i x_i+(1-\alpha_i)y_i\,,\;\; y'_i=\alpha_i y_i+(1-\alpha_i)x_i\,, \tag{7}$$

where α_i is a random number between 0 and 1.

Mutation is an additional operator executed after cross over in some of the individuals. A certain number of genes are randomly selected to change their values. This process is simulating biological genetic mutation. This is a preventive operation form trapping of the solution in local minimums.

In addition, the elitist strategy, in which the best solution of each generation is copied to the next generation, is applied to insure improvement of the best individual generation by generation.

5. Termination: The algorithm is stopped based on the maximum number of generations.

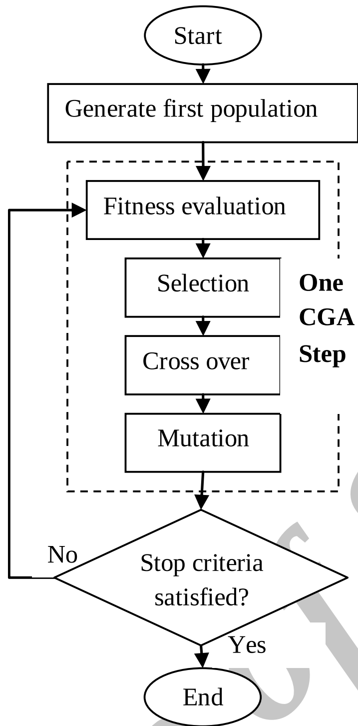


Figure 1. Flowchart of continuous genetic algorithm

5. THE PROPOSED STRATEGY: CGA-SBI-MS

In the proposed strategy, after each CGA step (i.e. fitness evaluation, selection, crossover and mutation) two sub-programs, SBI and MS, playing different roles, are executed. The flowchart of this method is outlined in Figure 2. Addition of these two sub-programs effectively specialize the standard CGA for damage detection.

These two subprograms are as follows:

1) Sensitivity based improvement (SBI): this sub-program improves individuals by using sensitivity analysis. To this end, for each individual of the generation its sensitivity matrix, **S**, is first established, then $\Delta\mathbf{X} = \mathbf{S}^{+}\Delta\mathbf{R}$ is added to it.
2) Micro search (MS): In this sub-program design variables of the optimization are reduced. After each *N2* generations of genetic algorithm the low variables of the best chromosome are set to zero and eliminated from optimization procedure.

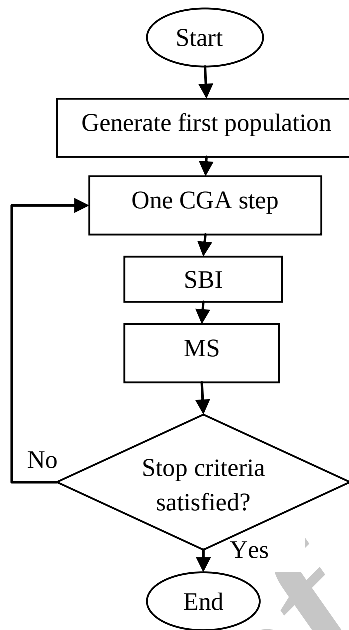


Figure 2. General flowchart of CGA-SBI-MS

It is noteworthy that; in usual damage detection problems just a few numbers of structural elements are damaged. In the other hand, it is expected after several generations of CGA the best individual of the population somewhat approaches the real damaging [4, 19], which is commonly a sparse vector. Therefore, the low variables of best individual can be considered as intact elements and be fixed to zero to eliminate from design variables and accelerate damage detection.

The step by step summary of the SBI sub-program process is as follows:

1. Predefine required values, $\Delta\mathbf{R}$, i and j. i and j are initially unity, however j may increase in MS part during the optimization process.
2. For each individual evaluate its sensitivity matrix, which is the derivatives of the structural responses with respect to structural damaging. For this purpose use the corresponding formulas presented in Section 3.
3. Improve each individual by adding $\Delta\mathbf{X} = \mathbf{S}^{+}\Delta\mathbf{R}$ to it. It is obvious that for each individual its corresponding sensitivity matrix must be used.
4. if i is equal to j end the sub-program and go to MS, otherwise; increase i one unit and repeat this sub-program from step one.

The step by step procedure of micro search (MS) is as follows:

1. Predefine these values: *N2*, *Iter*, *b*, a.

N2 is an integer number (e.g. 10), *Iter* is initially unity, but it changes during the optimization process, *b* is a small value (e.g. 0.1), where the elements with damage ratio lower than *b* are assumed to be undamaged, and a is a constant equal to zero or one.

If *Iter* is not equal to *N*2 increase *Iter* one unit and end the sub-program, otherwise; continue the following steps.

2. Set *Iter*=1 and increase *j* by the amount of a.
3. Find $x_i \leq b$ in the best chromosome. Fix the corresponding variables in the current population to zero, and eliminate them from the proceedings of optimization process.

Choosing a= 0 leads *j* remains unity during optimization process. Therefore, the condition in step 4 of SBI is always satisfied and i.e. the step 4 of SBI can be omitted. However, choosing a=1 leads to repetition of $\Delta\mathbf{X} = \mathbf{S}^{+}\Delta\mathbf{R}$ in SBI part, which causes rapid convergence, but more computational time in each step.

The parameters a, *b* and *N2* may be adjusted to improve CGA-SBI-MS performance. For more description, the detailed flowchart of the proposed strategy is presented in Figure 3.

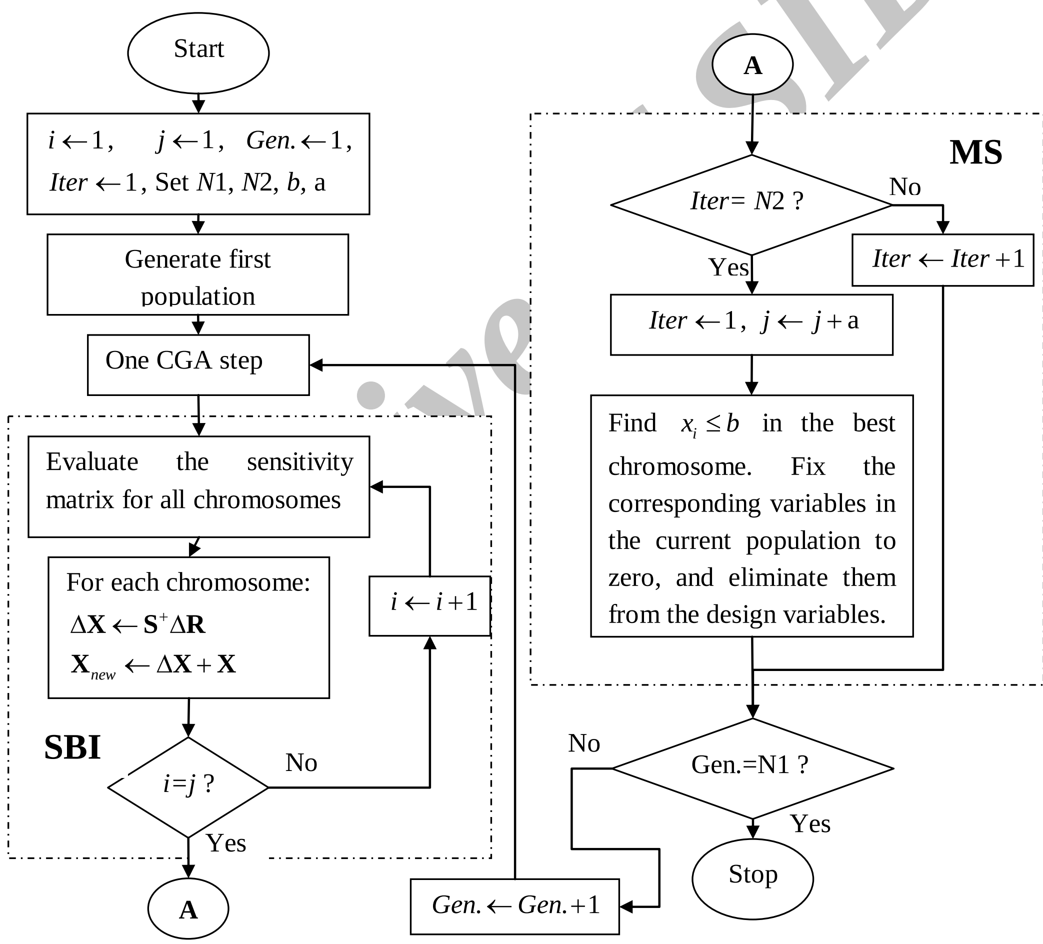


Figure 3. Detailed flowchart of CGA-SBI-MS

6. CASE STUDIES

In this part, the proposed method is applied to three different types of problems. These problems which all have been previously studied by other researchers are summarized in bellow:

1) Damage detection in a cantilever beam using natural frequencies. Damages are modeled as reduction in Young modulus of elements.
2) Multiple crack detection in a cantilever beam using natural frequencies. The cracks are modeled as a hinge and rotational springs in nodes.
3) Damage detection in a truss using static displacements affected by applied loads. Here, damages are also modeled as reduction in Young modulus of the elements.

Damage detection optimization problems usually have many local minimums, so there are different solution points with approximately the same structural response. This is the reason of defining a new index, referred to as *error in detection*, $ED = \|\mathbf{X} - \mathbf{X}_0\| = \sqrt{\sum_{i=1}^{nv} (x_i - x_{id})^2}$, where nv, x_i and x_{id} are the number of design variables, *i*th design variable of the solution point and *i*th design variable in the real damaged state, respectively. In all following case studies, for genetic algorithm, the population size is set to 50, the crossover probability is set to 0.8, and the mutation probability is set to 0.015.

6.1 Case study 1: A fifteen-element cantilever beam

A fifteen-element cantilever beam which was previously studied by Koh and Dyke [11] is considered to assess the efficiency of the proposed algorithm. The length, thickness and width of the beam are 2.74, 0.00635 and 0.0760 m, respectively and the elements are numbered from the fixed end as shown in Figure 4. In Ref. [11] the elements 4 and 12 have been assumed to be damaged by the extent of 30% and the process of damage detection has been proceed using the first five frequencies with three different ways from which the solution of genetic algorithm with the MDLAC criterion has been led to the best result. Here, the best result of Koh and Dyke is selected to be compared with those obtained by the proposed algorithm. For this case study, the parameters *b* and *N2* in the presented algorithm are set to 0.1 and 9, respectively.

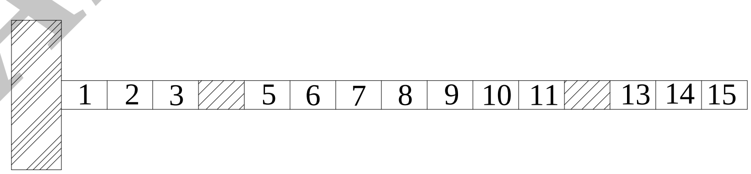
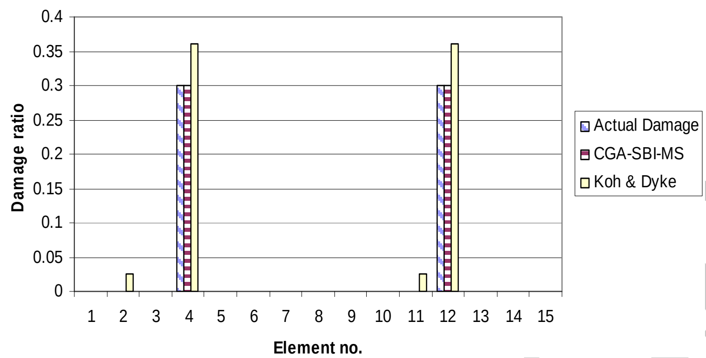


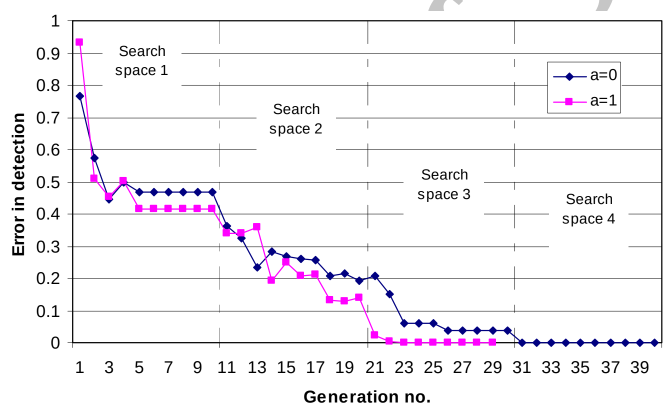
Figure 4. Cantilever beam having 15 elements, two damages are located in elements 4 and 12

The bar chart in Figure 5-a represents the final result of the proposed algorithm. As it can be seen, unlike Koh and Dyke method, the proposed algorithm is led to the exact solution.

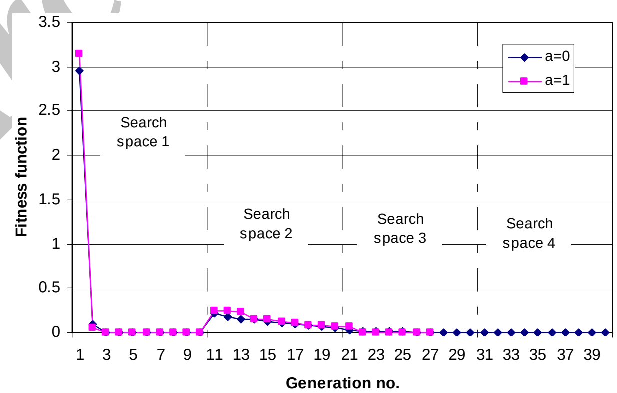
The convergence histories of the error in detection and the fitness function in both states of a=0 and a=1 are illustrated in Figure 5-b and Figure 5-c, respectively. As it is expected, when a=1, it is led to more rapid convergence. The design variables used during optimization process are listed in Table 1.



(a)



(b)



(c)

Figure 5. Solution results for 15 element cantilever beam: (a) Damage identification result (b) Convergence history of error in detection (c) Convergence history of fitness function

Table 1: Design variables of CGA-SBI-MS during optimization process

Design variables	**CGA-SBI-MS**	
	a=0	a=1
Search space 1	1,2, …,15	1,2, …,15
Search space 2	2 4 5 11 12 14 15	4 5 8 11 12 14 15
Search space 3	4 12 15	4 12
Search space 4	4 12	4 12

As it is shown in Figure 6, various combinations of CGA with SBI and MS sub-programs are investigated. The results shows, only CGA-SBI-MS is led to the exact solution, therefore in order to reach the exact solution by means of CGA, simultaneous utilization of both SBI and MS is inevitable.

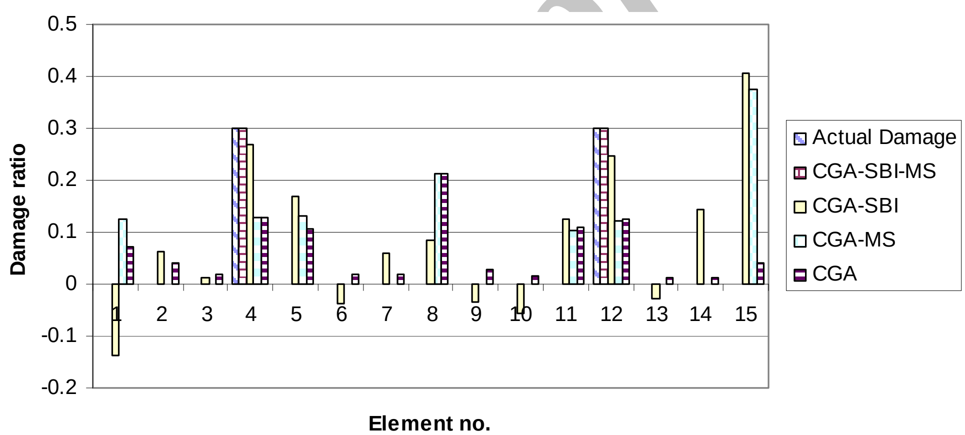


Figure 6. Comparison between various combinations of SBI and MS with CGA

6.2 Case study 2: A multiple cracked beam

6.2.1 Beam-spring model: A modeling of cracked beam

When a crack is occurred in a beam continuity of slops in the sides of the crack does not hold. Therefore, in a beam modeled by several beam elements, if the *i*th node has a crack it will have three degrees of freedom: deflection, right and left rotations $(w_i, \theta_i^L, \theta_i^R)$, otherwise two degrees of freedom: deflection and rotation (w_i, θ_i). The rotations θ_i^R and θ_i^L are related through a rotational spring with stiffness matrix as follows:

$$\mathbf{K}_c = \begin{bmatrix} K_t & -K_t \\ -K_t & K_t \end{bmatrix}, \tag{8}$$

where K_t, the torsional stiffness per unit width of the crack, is given by Nandwann and Maiti relation [13] as:

$$K_t = \frac{h^2 E}{72\pi\alpha^2 f(\alpha)}, \tag{9-a}$$

$$f(\alpha) = 0.6384 - 1.035\alpha + 3.7201\alpha^2 - 5.1773\alpha^3 + 7.553\alpha^4 - 7.332\alpha^5 + 2.4909\alpha^6, \tag{9-b}$$

where h is the height of the cross section and α is the ratio of crack depth to the height of the cross section.

Consider a cantilever beam produced by n same size beam elements with length of l connected by n rotational springs and hinges as shown in Figure 7. This cantilever beam is a model of the beam with n cracks locating in 0, l, 2l, 3l, …, (n-1)l positions from the fixed end. However, each of the cracks can be omitted by approaching the stiffness of its corresponding spring infinity. So, this beam-spring model can be used for modeling $1 \le i \le n$ cracks locating in some of 0, l, 2l, 3l, …, (n-1)l positions, by getting very large magnitude for the spring stiffness associated with the intact nodes.

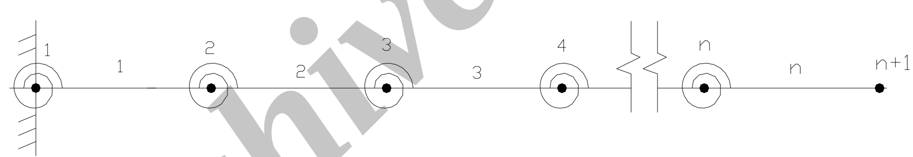


Figure 7. The beam-spring structure for modeling multiple cracks

The elemental displacement vector of the ith ($i \neq n$) and the nth elements are $\mathbf{W}_i^e = \{w_i, \theta_i^L, \theta_i^R, w_{i+1}, \theta_{i+1}^L, \theta_{i+1}^R\}^T$ and $\mathbf{W}_i^e = \{w_i, \theta_i^L, \theta_i^R, w_{i+1}, \theta_i\}^T$, respectively. The mass and stiffness matrices of the elements are given in the Appendix II.

6.2.2 A multiple cracked cantilever beam

The proposed strategy is experienced on a multiple cracked cantilever beam which was previously investigated by Lee [13]. The length and the height of the beam are L = 0.5 *m*, h = 0.02 *m*. Material Young's modulus and the density are E = 210 *GPa* and $\rho = 7860\,Kg/m^3$, respectively. In the present work, this beam is modeled with ten beam elements which are connected with hinges and rotational springs as it is expressed in previous and showed in Figure 7.

Two values $PL = \log_{10}(\lambda^*_{max} / \lambda^*_{min})$ and $NR = \sqrt{(\lambda_1 - \lambda_1^I)^2 + (\lambda_2 - \lambda_2^I)^2 + ...(\lambda_4 - \lambda_4^I)^2}$ are defined, where λ_i's , λ_i^I's and λ_i^*'s are the eigen-values (square of frequencies) of beam-spring, eigen-values of intact beam and the eigen-values of stiffness matrix of beam-spring structure, respectively. The value *PL* represents the conditioning of stiffness matrix, i.e. the lower *PL* means the better conditioning [20]. The value *NR* represents the difference in the structural behavior of the intact and the beam-spring model. Logically, it is expected when the stiffness of rotational springs approach infinity the structural natural frequencies equalize the intact ones and the value NR approach zero. However, in practice numerical problems prevent it. To represent this fact, the stiffness of all rotational springs are simultaneously increased to verify the *PL* and *NR* values. As it can be observed from Figure 8, because of conditioning problems, the values higher than $\log_{10} K_t$ =11.5 in the horizontal axis, disturb the *NR* diagram. Therefore, we set $K_t = 1e11 Nm$ as the upper bound of rotational springs' stiffness to prevent ill-conditioning. To verify the influence of this upper bound, the first four frequencies of intact beam and the beam-spring model with springs' stiffness of $K_t = 1e11 Nm$ are compared in Table 2. As it can be seen from the results, the beam-spring model can estimate the frequencies of intact beam until three digits after point with reasonable approximation.

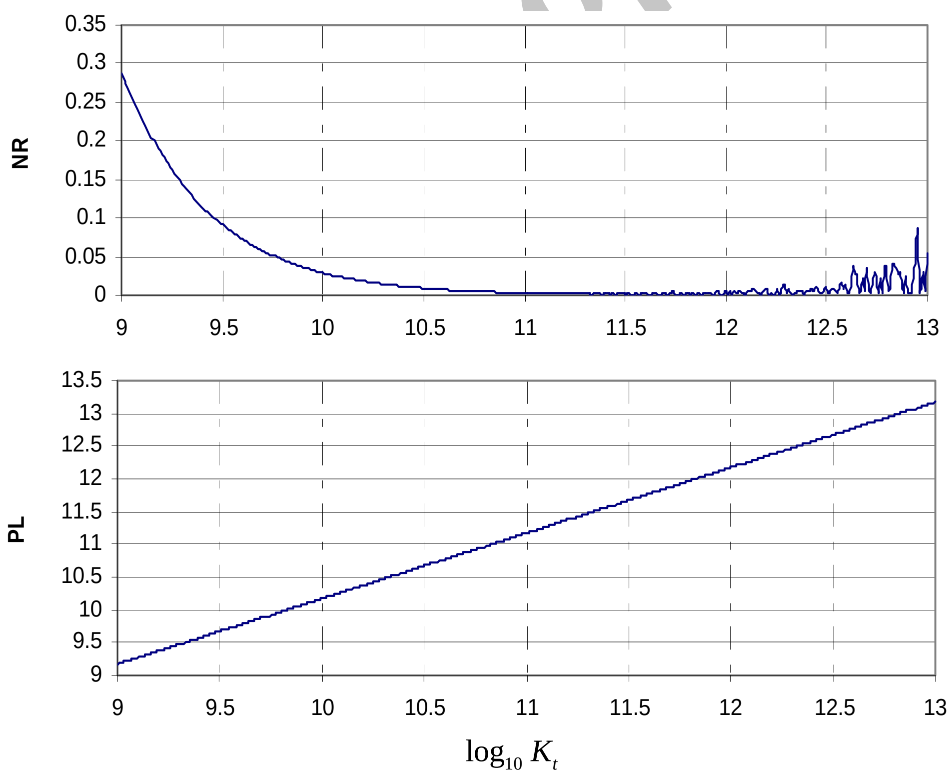


Figure 8. The effect of stiffness of rotational springs in numerical results

Table 2: Comparison of first four frequencies of real intact and beam-spring intact model

Intact models	Frq. 1	Frq. 2	Frq. 3	Frq. 4
Real intact model	66.497	405.218	1086.367	2004.819
Beam-spring intact model	66.497	405.218	1086.365	2004.817

The depth and location of the *i*th crack is represented by a_i and s_i, respectively (Figure 9).

The parameters $\alpha_i = a_i / h$ and $\beta_i = s_i / L$, are the normalized size and the normalized location of the *i*th crack, respectively, where *h* and *L* are the beam height and length, respectively.

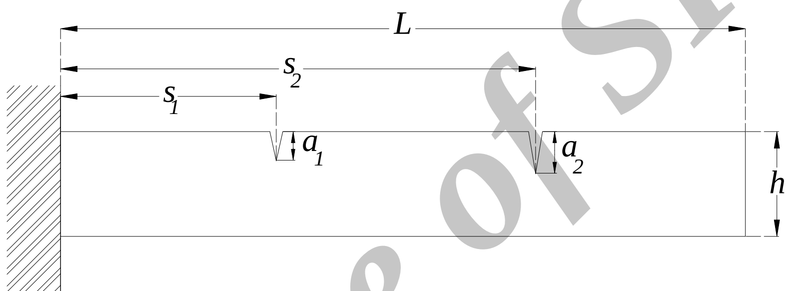


Figure 9. Cantilever beam with double cracks

For this case study, the parameters *b* and *N2* in CGA-SBI-MS are set to 0.05 and 10, respectively. Table 3 demonstrates the results of a scenario with three damages. As it can be seen, the proposed algorithm by using 2 frequencies lower than Lee's method leads to much better results. In the Lee's method the number of required frequencies for damage detection is a function of the number of damages. The results of another scenario with two damages is presented in Table 4. In the Lee's method the crack locations are considered as continuous variables. This cause – as it can be observed from Tables 3 and 4 – the crack locations in the Lee's results become inaccurate. Error in detection is evaluated from $ED = \sqrt{\sum_{i=1}^{nd} (\alpha_i - \alpha_{di})^2}$,

where α_i, α_{id} and *nd* are the normalized crack depths of solution points, normalized crack depth of the real damaged structure and number of damages, respectively. So, for evaluation of error in detection of Lee's method, the error of crack locations is ignored. Error in detection using CGA-SBI-SM in the states of a=0 and a=1 are given in Table 5. A general comparison between CGA-SBI-SM and Lee's method is presented in Table 6. The convergence histories of error in detection and fitness function just for the first scenario in the state of a=0 is presented in Figure 10. Here, for convergence history of error in detection, we just consider the error exposed from crack depth.

Table 3: Comparison of detected cracks between CGA-SBI-MS and Lee's work for scenario1

Scenario 1	**Utilized Info.**	α_1	α_2	α_3	β_1	β_2	β_3
Actual damage		0.1	0.1	0.1	0.2	0.4	0.8
Lee (after 20 iterations)	6 Frqs.	0.1011	0.1024	0.0975	0.2094	0.401	0.8076
CGA-SBI-MS (a=0)	4 Frqs.	0.0997	0.1006	0.0996	0.2	0.4	0.8
CGA-SBI-MS (a=1)	4 Frqs.	0.0998	0.1005	0.0997	0.2	0.4	0.8

Table 4: Comparison of detected cracks between CGA-SBI-MS and Lee's work for scenario2

Scenario 2	**Utilized Info.**	α_1	α_2	β_1	β_2
Actual damage		0.2	0.2	0.4	0.6
Lee (after 10 iterations)	4 Frqs.	0.2029	0.1971	0.4028	0.6028
CGA-SBI-MS (a=0)	4 Frqs.	0.20012	0.1999	0.4	0.6
CGA-SBI-MS (a=1)	4 Frqs.	0.19999	0.20001	0.4	0.6

Table 5: Comparison of error in detection between CGA-SBI-MS and Lee's work

Error in detection	**Lee**	**CGA-SBI-MS (a=0)**	**CGA-SBI-MS (a=1)**
Scenario 1	0.003636	0.000729686	0.000615362
Scenario 2	0.004101	0.000156205	1.41421E-05

Table 6: Comparison of CGA-SBI-MS for beam-spring model with Lee's method

Comparison of the proposed algorithm with respect to Lee's	
Advantages:	Disadvantage:
More accurate	Treats the cracks locations as discrete variables
The required information is independent of no. of damages	

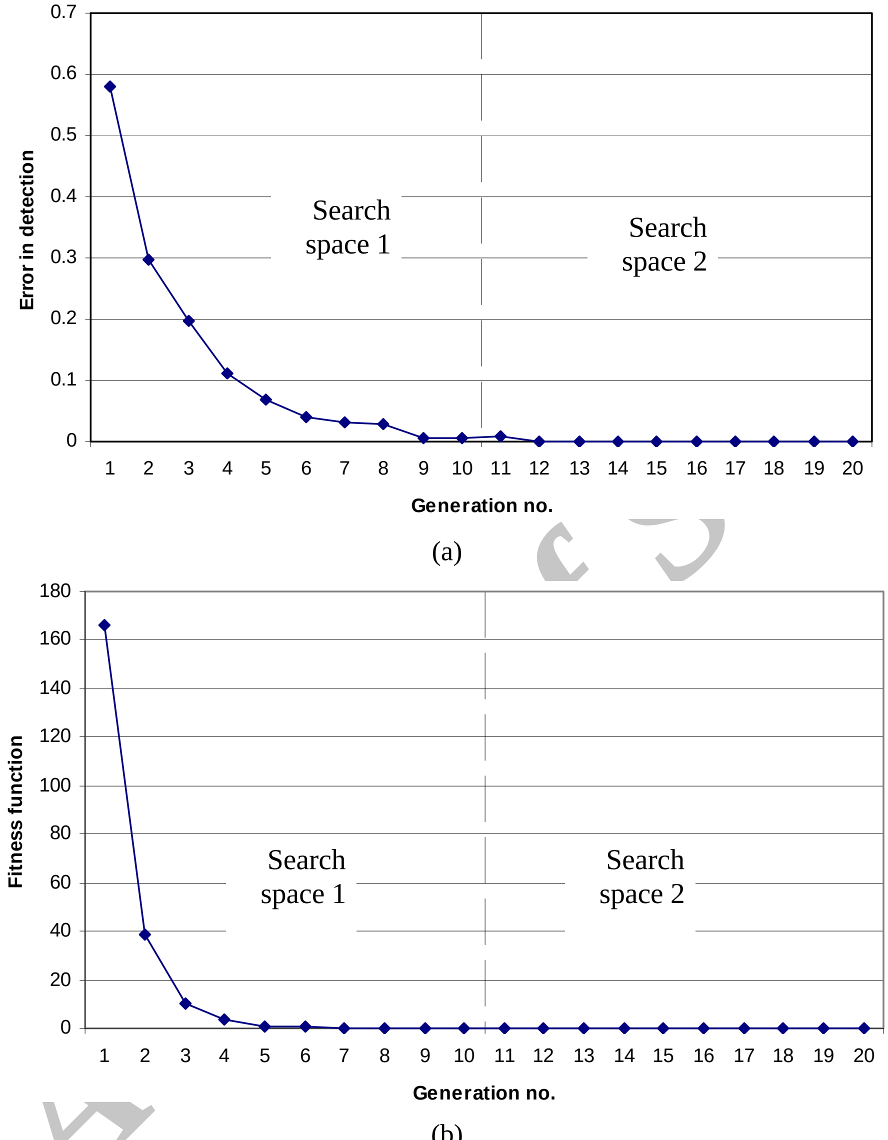


Figure 10. Solution results of crack detection obtained by CGA-SBI-MS (a=0) for scenario1 of the cantilever beam: (a) Convergence history of error in detection (b) Convergence history of fitness function

6.3 Case study 3: A planar truss

A two-dimensional truss, shown in Figure 11, has been previously studied by Bakhtiari-Nejad et al. [21]. Here, those case studies are used to assess the ability of CGA-SBI-MS in static cases.

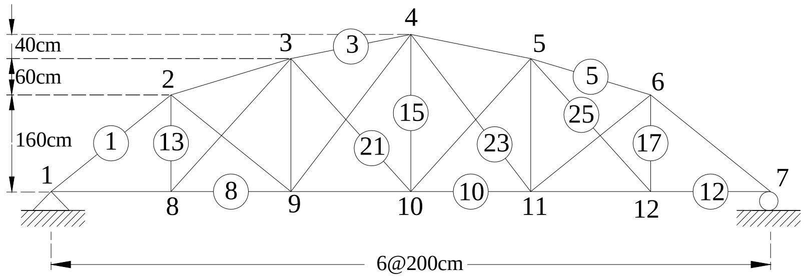
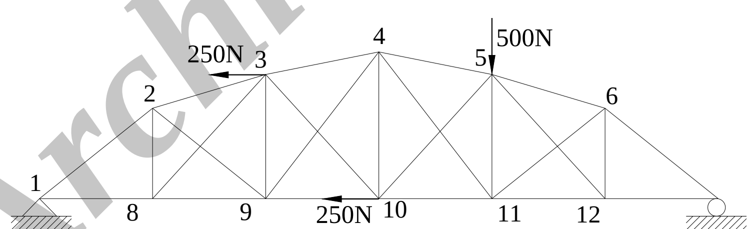


Figure 11. Geometry of bowstring truss

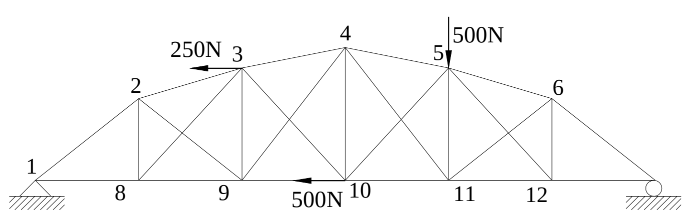
Bakhtiari-Nejad et al. judiciously select the positions of applied loads and measurement locations to facilitate damage detection. Here, the load cases and measurement locations are adopted from their work as it is shown in Figure 12 and Figure 13, respectively. Axial rigidity of truss members is given in Table 7.

Table 7: Axial rigidity of bowstring truss members

Member	Axial rigidity (kN)
1-6	3.6e5
7-12	3e5
13-17	2e5
18-25	2.4e5



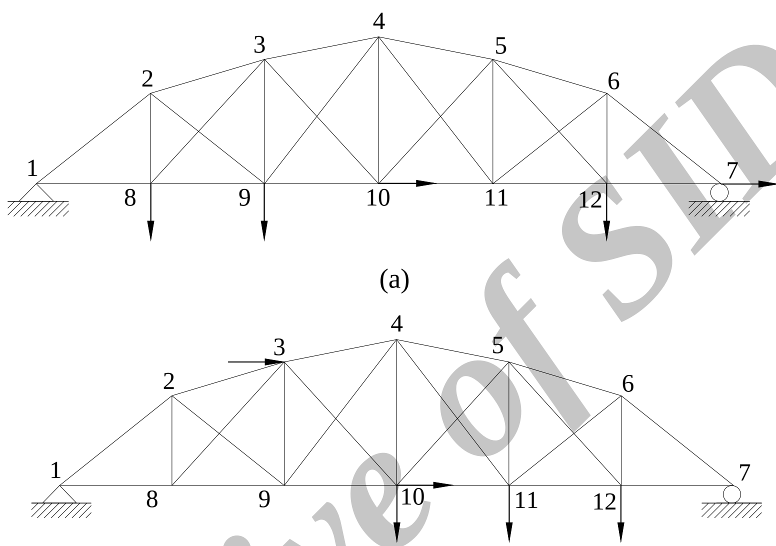
(a)



(b)

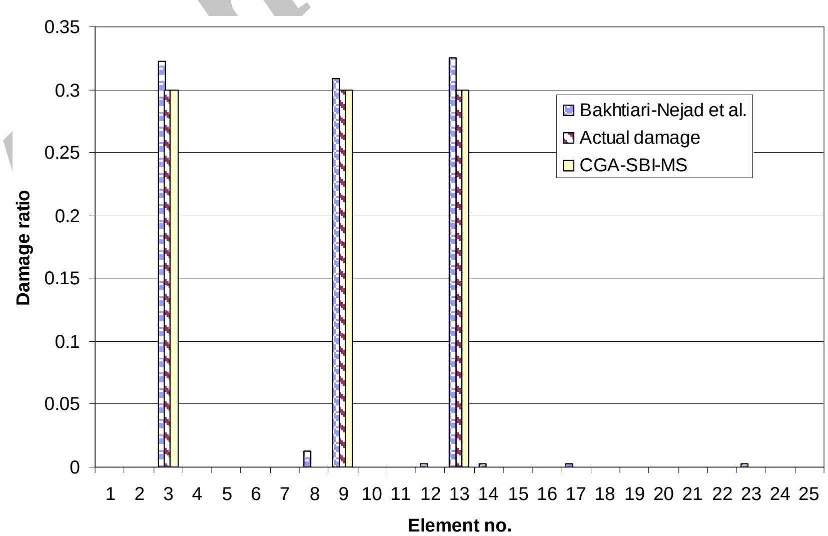
Figure 12. Applied loading (a) load case 1 (b) load case 2

For this case study, the parameters *b* and *N2* in the present algorithm are set to 0.1 and 9, respectively. Here, two scenarios from Bakhtiari-Nejad et al. work are considered. The damage identification results and the convergence histories of error in detection and fitness function of CGA-SBI-MS (a=0) for both scenarios are presented. As it can be observed from the bar charts, unlike Bakhtiari-Nejad et al.'s method, CGA-SBI-MS is led to the exact solution. Therefore, it has been realized that the proposed algorithm can also be utilized in damage detection by static measurements.

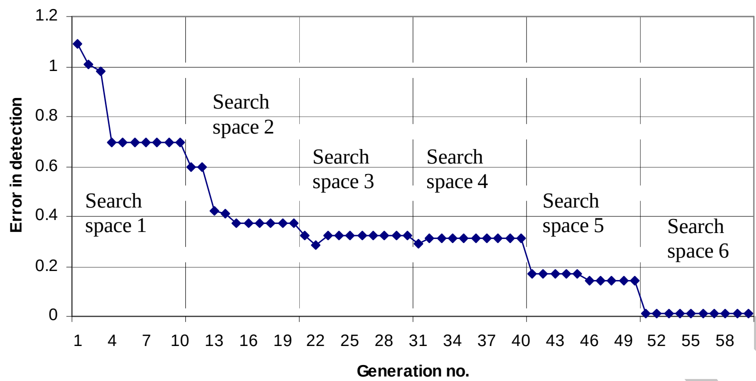


(a) (b)

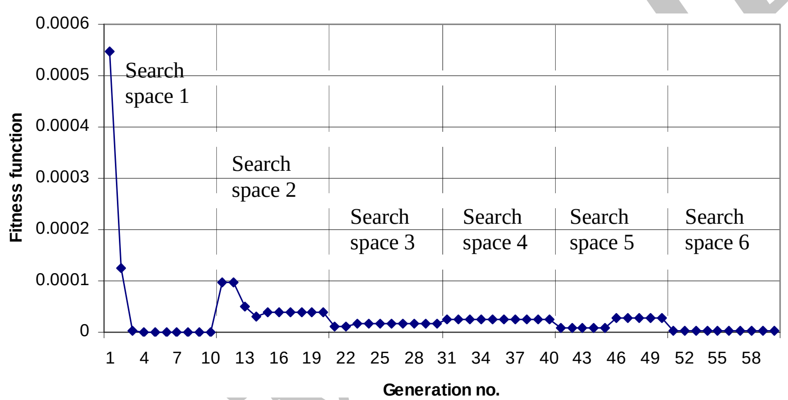
Figure 13. Measurement locations (a) load case1 (b) load case 2



(a)

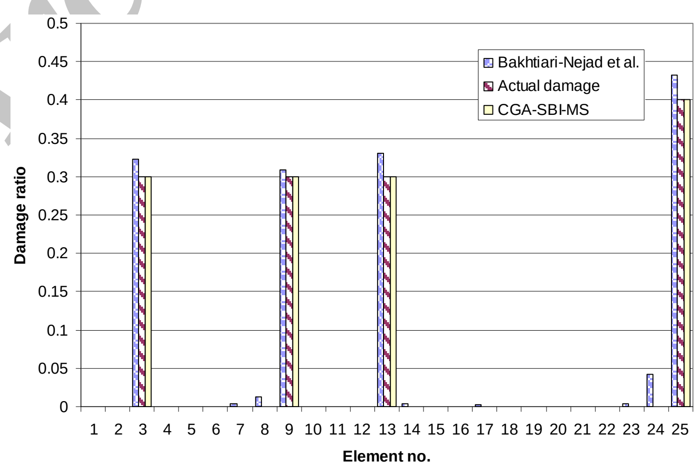


(b)

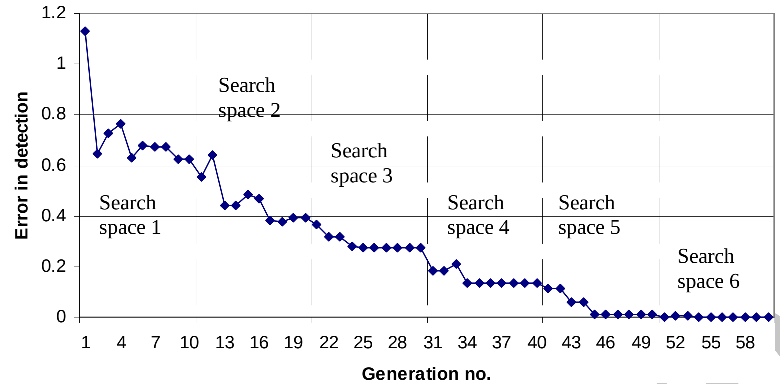


(c)

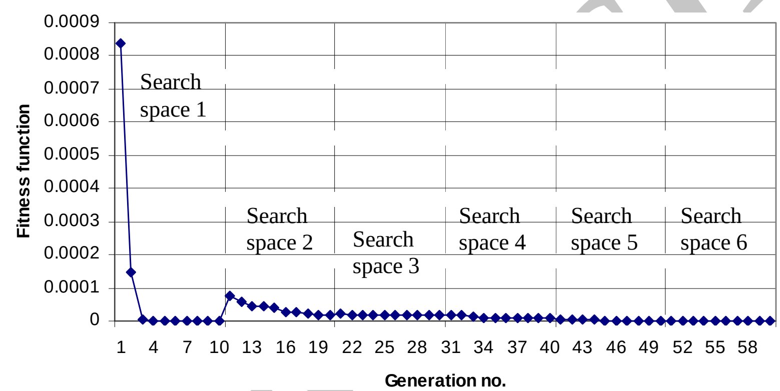
Figure 14. Solution results obtained by CGA-SBI-MS(a=0) for scenario 1 of the planar truss: (a) Damage identification result (b) Convergence history of error in detection (c) Convergence history of fitness function



(a)



(b)



(c)

Figure 15. Solution results obtained by CGA-SBI-MS(a=0) for scenario 2 of the planar truss: (a) Damage identification result (b) Convergence history of error in detection (c) Convergence history of fitness function

7. CONCLUSION

Damage detection problems are equivalent to a nonlinear system of equations in which practically the number of equations is less than the number of unknowns. Such problems have infinite solutions, however a solution with low l_0 and l_2 norms is the right one. In this paper, the conventional CGA is developed to be specialized for damage detection. This development is done by adding two sub-programs after typical CGA reproduction process. The first sub-program, SBI, improves the individuals depart from the CGA's reproduction by solving the linearized problem about each individual using Moore-Penrose pseudo inverse. The second sub-program is reducing the search space after each several generations by elimination of low variables of the best individual. This strategy is applied to three different type case studies, and in all states its merit has been proved.

APPENDIX I

Matrix **S**, can be factorized as $\mathbf{S}_{m\times n} = \mathbf{U}_{m\times m}\mathbf{\Sigma}_{m\times n}\mathbf{V}_{n\times n}$ (singular value decomposition of **S**), where the columns of **V** (right singular vectors) are the eigenvectors of $\mathbf{S}^T\mathbf{S}$, the columns of **U** (left singular vectors) are the eigenvectors of $\mathbf{S}\mathbf{S}^T$ and $\mathbf{\Sigma}_{m\times n}$ is a diagonal matrix in which $\mathbf{\Sigma}_{ii}$ s (the singular values of **S**) are the square roots of the eigenvalues of $\mathbf{S}\mathbf{S}^T$ and $\mathbf{S}^T\mathbf{S}$ that correspond with the columns in **U** and **V**. The pseudo inverse of **S** can be calculated by $\mathbf{S}^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}$, where the matrix $\mathbf{\Sigma}^+$ is created by replacing every nonzero entry of $\mathbf{\Sigma}$ by its reciprocal and transposing the resulting matrix.

APPENDIX II

In the beam-spring model with *n* elements, the stiffness matrix of the Euler–Bernoulli beam elements are changed from its typical state presented in Eq. (I) to Eq. (II-a) for *i*th element ($i \neq n$) and to Eq. (II-b) for *n*th element. The mass matrix of the beam elements for *i*th element ($i \neq n$) and *n*th element are presented in Eqs. (III-a) and (III-b), respectively.

$$\mathbf{K}_e = \frac{EI}{l^3}\begin{bmatrix} 12 & 6l & -12 & 6l \\ & 4l^2 & -6l & 2l^2 \\ & & 12 & -6l \\ & & & 4l^2 \end{bmatrix}, \quad \text{(I)}$$

$$\underset{i\neq n}{\mathbf{K}_e^i} = \frac{EI}{l^3}\begin{bmatrix} 12 & 0 & 6l & -12 & 6l & 0 \\ & 0 & 0 & 0 & 0 & 0 \\ & & 4l^2 & -6l & 2l^2 & 0 \\ & & & 12 & -6l & 0 \\ & & & & 4l^2 & 0 \\ & & & & & 0 \end{bmatrix} \text{(II-a)}, \quad \mathbf{K}_e^n = \frac{EI}{l}\begin{bmatrix} 12 & 0 & 6l & -12 & 6l \\ & 0 & 0 & 0 & 0 \\ & & 4l^2 & -6l & 2l^2 \\ & & & 12 & -6l \\ & & & & 4l^2 \end{bmatrix}, \text{(II-b)}$$

$$\underset{i \neq n}{\mathbf{M}_e^i} = \frac{\rho A l}{420}\begin{bmatrix} 156 & 0 & 22l & 54 & -13l & 0 \\ & 0 & 0 & 0 & 0 & 0 \\ & & 4l^2 & 13l & -3l^2 & 0 \\ & & & 156 & -22l & 0 \\ & & & & 4l^2 & 0 \\ & & & & & 0 \end{bmatrix}, \qquad \text{(III-a)}$$

$$\mathbf{M}_e^n = \frac{\rho A l}{420}\begin{bmatrix} 156 & 0 & 22l & 54 & -13l \\ & 0 & 0 & 0 & 0 \\ & & 4l^2 & 13l & -3l^2 \\ & & & 156 & -22l \\ & & & & 4l^2 \end{bmatrix}. \qquad \text{(III-b)}$$

REFERENCES

1. Tsaig Y, Donoho DL. Breakdown of equivalence between the minimal l^1-norm solution and the sparsest solution, *Signal Processing*, **86**(2006) 533-48.
2. Koh CG, Perry MJ. *Structural Identification and Damage Detection Using Genetic Algorithm*, CRC Press, Taylor and Francis Group, London, UK, 2010.
3. Au FTK, Cheng YS, Tham LG, Bai ZZ. Structural damage detection based on a Micro-genetic algorithm using incomplete and noisy test data, *Journal of Sound and Vibration*, No. 5, **259**(2003) 1081-94.
4. Guo HY, Li ZL. A two-stage method to identify structural damage sites and extents by using evidence theory and micro-search genetic algorithm, *Mechanical Systems and Signal Processing*, **23**(2009) 769-82.
5. He RS, Hwang SF. Damage detection by an adaptive real-parameter simulated annealing genetic algorithm under the assistance of grey relation analysis, *Computers and Structures*, **84**(2006) 2231-43.
6. Wang X, Hu N, Fukunaga H, Yao ZH. Structural damage identification using static test data and changes in frequencies, *Engineering Structures*, **23**(2001) 610-21.
7. Perera R, Ruiz A. A multistage FE updating procedure for damage identification in large-scale structures based on multiobjective evolutionary optimization, *Mechanical Systems and Signal Processing*, **22**(2008) 970-91.
8. Kokot S Zembaty Z, Damage reconstruction of 3D frames using genetic algorithms with Levenberg–Marquardt local search, *Soil Dynamics and Earthquake Engineering*, **29**(2009) 311-23.
9. Sahoo B, Maity D. Damage assessment of structures using hybrid neuro-genetic algorithm, *Applied Soft Computing*, **7**(2007) 89-104.
10. Messina A, Williams EJ, Contursi T. Structural damage detection by a sensitivity and statistical-based method, *Journal of Sound and Vibration*, No. 5, **216**(1998) 791-808.

11. Koh BH, Dyke SJ. Structural health monitoring for flexible bridge structures using correlation and sensitivity of modal data, *Computers and Structures*, **85**(2007) 117-30.
12. Gomes HM, Silva NRS. Some comparisons for damage detection on structures using genetic algorithms and modal sensitivity method, *Applied Mathematical Modelling*, **32**(2008) 2216-32.
13. Lee J. Identification of multiple cracks in a beam using natural frequencies, *Journal of Sound and Vibration*, **320**(2009) 482-90.
14. Vakil-Baghmisheh MT, Peimani M, Homayoun Sadeghi M, Ettefagh MM. Crack detection in beam-like structures using genetic algorithms, *Applied Soft Computing*, **8**(2008) 1150-60.
15. Caddemi S, Morassi A. Crack detection in elastic beams by static measurements, *International Journal of Solids and Structures*, **44**(2007) 5301-15.
16. Khaji N, Shafiei M, Jalalpour M. Closed-form solutions for crack detection problem of Timoshenko beams with various boundary conditions, *International Journal of Mechanical Sciences*, **51**(2009) 667-81.
17. Wittrick WH. Rates of change of eigenvalues, with reference to buckling and vibration problems, *Journal of the Royal Aeronautical Society*, **66**(1962) 590-1.
18. Rahamia H, Kaveh A, Gholipour Y. Sizing, geometry and topology optimization of trusses via force method and genetic algorithm, *Engineering Structures*, **30**(2008) 2360-9.
19. Salajegheh E, Gholizadeh S. Optimum design of structures by an improved genetic algorithm using neural networks, *Advances in Engineering Software*, **36**(2005) 757-67.
20. Kaveh A, Ghaderi J. Conditioning of stiffness matrices of structures, *Computers and Structures*, No. 4, **63**(1997) 719-25.
21. Bakhtiari-Nejad F, Rahai A, Esfandiari A. A structural damage detection method using static noisy data, *Engineering Structures*, **27**(2005) 1784-93.