INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2012; 2(4):489-510

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018

A TWO-STAGE METHOD FOR DAMAGE DETECTION OF LARGE-SCALE STRUCTURES

S.S. Naseralavi[a], E. Salajegheh[*,†,a] J. Salajegheh[a] and M. Ziaee[b]
[a]*Department of Civil Engineering, University of Kerman, Kerman, Iran*
[b]*Department of Civil Engineering, Kerman Graduate University of Technology, Kerman, Iran*

ABSTRACT

A novel two-stage algorithm for detection of damages in large-scale structures under static loads is presented. The technique utilizes the vector of response change (VRC) and sensitivities of responses with respect to the elemental damage parameters (RSEs). It is shown that VRC approximately lies in the subspace spanned by RSEs corresponding to the damaged elements. The property is leveraged in the first stage of the proposed method by seeking RSEs whose spanned subspace best contains the VRC. Consequently, the corresponding elements are regarded as damage candidates. To alleviate the exploration among RSEs, they are first partitioned into several clusters. Subsequently, discrete ant colony optimization (ACO) is utilized to find the clusters containing the RSEs of damaged elements. In the second stage of the algorithm, damage amounts for the restricted elements are determined using a continuous version of ACO. Two numerical examples are studied. The results illustrate that the method is both robust and efficient for detection of damages in large-scale structures.

Received: 30 July 2012; Accepted: 30 September 2012

KEY WORDS: damage detection; large-scale structure; ant colony; sensitivity analysis; static analysis; system of equations

1. INTRODUCTION

Damage detection is one of the branches of structural health monitoring which has recently

[*] Corresponding author: E. Salajegheh, Department of Civil Engineering, University of Kerman, Kerman, Iran
[†]E-mail address: eysasala@mail.uk.ac.ir

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018

attracted many scientific efforts. Damage detection techniques have been successfully applied to several real-world problems. Recent work widely employ meta-heuristic searching tools for damage diagnosis. For example, Koh et al. [1] detected the damages using accelerations of a shear building under dynamic loads. To this aim, they improved the GA in two ways by embedding two different types of local search algorithms in the GA body. In a study, Koh and Dyke [2] detected damages of a cable stayed bridge by employing the GA for maximizing the response correlation for the actual and the hypothetically damaged structures. An application of the micro-genetic algorithm to detect a single crack in a real cracked beam has also been employed by Vakil-Baghmisheh et al. [3]. Begambre and Laier identified the structural damages by using FRFs and employing a hybridized particle swarm optimization with simplex algorithm [4]. Naseralavi et al. [5] improved the real coded genetic algorithm using sensitivity matrices of structural responses with respect to damage severities for diagnosing the damaged elements. Meruane and Heylen [6] also addressed an under-determined damage detection problem. To find the unique damage solution, they penalized the conventional objective function by summation of damage extents. They also hybridized the real-coded genetic algorithm through embedding a local search algorithm in the GA body to improve its performance for damage detection. Yu and Xu [7] employed continuous ACO to detect the damaged elements through minimizing the corresponding objective function. In their study, they used natural frequencies and mode shapes to access the damage. Many recently proposed methods consist of two stages for under-determined damage detection problems. In the first stage of these methods, the search space is usually reduced toward damaged elements. In the second stage, the true damage solution is computed by solving the corresponding optimization problem [8-12].

Static damage identification methods are usually simpler than the dynamic ones, since the static equilibrium equation is only relevant to the stiffness properties of structures. Moreover, the equipments for static testing are comparatively cheaper [12]. One advantage of using static responses is that they are more sensitive to damages than natural frequencies and mode shapes [13]. Conversely, static responses are less popular in the case of highly stiff structures due to implementation problems. Combinations of static response with other types of responses are utilized for damage diagnosis in the previous work [14,15]. Static responses are also merely employed for damage detection in Refs. [16-18].

This paper proposes a two-stage method specialized for damage detection of large-scale structures with several hundred members using static data. In the first stage, the search space is reduced through minimizing a new objective function using the discrete ACO. In the second stage, the continuous version of ACO is employed to identify the damage extents. To illustrate the efficiency of the proposed method, two large-scale steel trusses are studied. Finally, it is concluded that the method is promising in detection of damages in structures with hundreds of members.

2. STATIC DAMAGE DETECTION USING SENSITIVITY ANALYSIS

Due to damages, the stiffness of the damaged elements is reduced. Hence, the displacements under static loads are increased in comparison to the healthy state. The static equilibrium

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018

equation of the structure can be expressed as:

$$\mathbf{KU} = \mathbf{F} \tag{1}$$

where **K** is the stiffness matrix, **U** is the displacements vector, and **F** is the applied loads vector. By differentiating Eq. (1) with respect to a design variable (e.g. damage variable of an element), the sensitivity of structural displacements with respect to that design variable can be computed as [5]:

$$\mathbf{K}'\mathbf{U} + \mathbf{K}\mathbf{U}' = \mathbf{0} \Rightarrow \mathbf{U}' = -\mathbf{K}^{-1}\mathbf{K}'\mathbf{U} \tag{2}$$

For damage identification, one has to find the set of damage variables in a way that the analytical displacements of the structure would best fit the measured ones. Damage detection problems can be expressed mathematically as below [5]:

$$\mathbf{U}_d = \mathbf{U}(\mathbf{X}) \tag{3}$$

where $\mathbf{X} = (x_1\ ,\ x_2\ ,\ldots\ x_n)^{\mathrm{T}}$ is called the damage vector, and $0 \le x_i \le 1$ is the damage extent (damage ratio) of the *i*th element where $x_i = 0$ and $x_i = 1$ indicate the intact and completely damaged states, respectively; and *n* is the number of structural elements. $\mathbf{U}(\mathbf{X}) = (u_1(\mathbf{X})\ ,\ u_2(\mathbf{X})\ ,\ \ldots\ u_m(\mathbf{X}))^{\mathrm{T}}$ is the vector of *m* displacements of hypothetically damaged structure with damage state **X** that can be evaluated from the analytical model, and $\mathbf{U}_d = (u_{d,1}\ ,\ u_{d,2}\ ,\ \ldots\ u_{d,m})^{\mathrm{T}}$ is the vector of *m* structural displacements of the existing damaged structure. Eq. (3) can be estimated by using the first order approximation as follows [19]:

$$\mathbf{U}_d = \mathbf{U}(\mathbf{X}) = \mathbf{U}_h + \frac{\partial \mathbf{U}}{\partial \mathbf{X}}\Delta\mathbf{X} + \ldots \Rightarrow \overbrace{\mathbf{U}_d - \mathbf{U}_h}^{\Delta\mathbf{U}} \simeq \overbrace{\frac{\partial \mathbf{U}}{\partial \mathbf{X}}}^{\mathbf{S}}(\mathbf{X}_d - \overbrace{\mathbf{X}_h}^{\mathbf{0}_{n\times 1}}) \tag{4}$$
$$\therefore \mathbf{S}_{m\times n}.\mathbf{X}_{n\times 1} = \Delta\mathbf{U}_{m\times 1}$$

where $\mathbf{U}_h$ is the structural displacement vector of the healthy structure, $\Delta\mathbf{X}$ is the damage change vector caused by actual damage, $\mathbf{S} = \partial\mathbf{U}/\partial\mathbf{X}$ is the sensitivity matrix, and $\mathbf{X}_d = \left(x_{d,1}, x_{d,2}, \ldots, x_{d,n}\right)^T$ is the actual damage vector. Notice that $\mathbf{U}(\mathbf{0}) = \mathbf{U}_h$ and $\mathbf{U}\left(\mathbf{X}_d\right) = \mathbf{U}_d$. $\Delta\mathbf{U} = \mathbf{U}_d - \mathbf{U}_h$ is the vector of the structural response change due to the damage. We denote the *i*th column of **S** as $\mathbf{S}_i$, and hence $\mathbf{S} = \left[\mathbf{S}_1\ \mathbf{S}_2 \ldots \mathbf{S}_n\right]$. To ease the explanations, we also refer to $\mathbf{S}_i$'s and $\Delta\mathbf{U}$ as response sensitivities to elements (RSEs) and the vector of response change (VRC), respectively. Considering Eq. (2), the *i*th column of matrix **S**, i.e. $\mathbf{S}_i = \partial\mathbf{U}/\partial x_i$, can be computed as $-\mathbf{K}^{-1}(\partial\mathbf{K}/\partial x_i)\mathbf{U}$.

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018

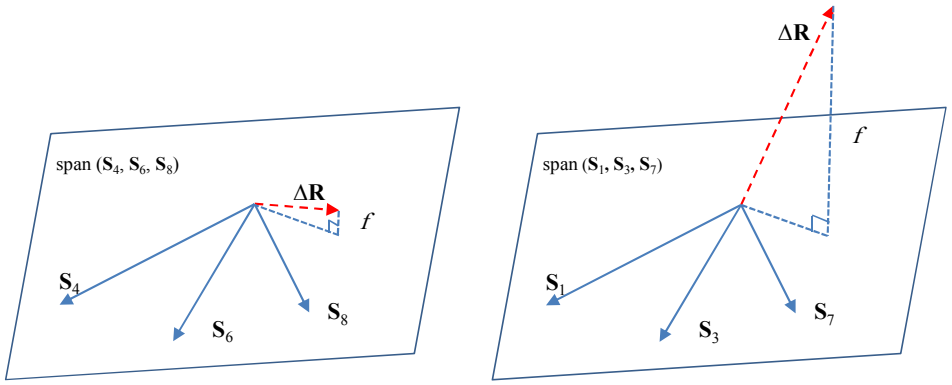
3. THE PROPOSED ALGORITHM

3.1. The first stage

3.1.1. The objective function

Due to the fact that in static response the number of measurements is usually less than the number of structural members, so the equation $\Delta\mathbf{U} = \mathbf{S}.\mathbf{X}$ is under-determined ($n > m$), and hence has infinite solutions. Mathematically speaking, there is an $(n-m)$-dimensional subspace in R^n that its points are solutions of $\Delta\mathbf{U} = \mathbf{S}.\mathbf{X}$. To find the true unique damage solution within the subspace, it should be noted that in the damaged structures most of the structural elements are still intact and, therefore, the true solution has high sparsity. In the first stage of the algorithm, we are to restrict the damage candidates by considering some of the elements healthy and eliminating them from the design variables. Since $\Delta\mathbf{U} \simeq \mathbf{S}.\mathbf{X}_d = \sum_{i=1}^{n} x_{d,i}\mathbf{S}_i$, and the damage ratio of the undamaged elements are zero, thus we can write $\Delta\mathbf{U} \simeq \sum_{i \in SDE} x_{d,i}\mathbf{S}_i$, where *SDE* is the set of damaged elements. This means $\Delta\mathbf{U}$ lies approximately in the span of such $\mathbf{S}_i$'s corresponding to the damaged elements, i.e., $\Delta\mathbf{U}$ lies approximately in span$\{\mathbf{S}_i : i \in SDE\}$. In other words, VRC lies approximately in the subspace spanned by RSEs of the damaged elements. Therefore, it is expected that the component of $\Delta\mathbf{U}$ orthogonal to span$\{\mathbf{S}_i : i \in SDE\}$ is smaller in magnitude than orthogonal component of $\Delta\mathbf{U}$ with respect to other same dimensional subspaces spanned by other combinations of $\mathbf{S}_i$'s [20]. Here, the same dimension condition for such a comparison is imposed due to the fact that by enlarging a subspace through adding new dimensions to it, the orthogonal component of a certain vector to that subspace will get shortened. The component of $\Delta\mathbf{U}$ along $\mathbf{S}_i$; $i \in SDE$ is approximately equal to the damage ratio of the *i*th element. Let us review the above discussions with an example for better understanding. Consider a structure in which the elements 4, 6, and 8 are damaged. Therefore, as what is schematically shown in Figure 1, $\Delta\mathbf{U}$ lies approximately in the subspace span$(\mathbf{S}_4, \mathbf{S}_6, \mathbf{S}_8)$, i.e., the orthogonal vector $\Delta\mathbf{U}_{\perp \mathrm{span}(\mathbf{S}_4,\mathbf{S}_6,\mathbf{S}_8)}$ which is denoted by *f* in the figure has a very small magnitude. Meanwhile, the orthogonal component of $\Delta\mathbf{U}$ to any other 3-dimensional subspaces spanned by any other 3-combinations of $\mathbf{S}_i$'s is not generally that small as shown in Figure 1 (b).

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018



(a) (b)

Figure 1. Orthogonal component of $\Delta\mathbf{U}$ with respect to: (a) the subspace associated with RSEs of all damaged elements, and (b) the subspace which is not associated with RSEs of the damaged elements

As aforementioned, to find the damaged elements, one might excessively search for such RSEs so that VRC best lies in the subspace spanned by them, and consequently take the corresponding elements as damaged ones. However, such a kind of approach is computationally expensive and therefore impractical for large-scale structures. This idea is adopted in the first stage for restricting the potentially damaged elements with two stratagems for alleviating the searching effort: First, the elements are clustered into several groups and then damaged clusters (clusters consisting damaged elements) are searched instead of damaged elements themselves to reduce the computational effort. To this aim, searching is done for some clusters so that VRC best lies in the subspace spanned by their RSEs. Second, instead of excessively searching among damage clusters, discrete ACO is employed for the searching process.

Let us elaborate on the clustering effect by an example. Consider a 1000-element structure having five damaged elements. In the case of no clustering, we have to search among $\binom{1000}{5} = 8.2503\text{e+}012$ items to find the true damage members. By clustering the elements to four member groups as an instance, the number of clusters will be $1000/4 = 225$. It should be noted that if the number of elements was not dividable into the number of clusters, we would duplicate some of the elements to make it dividable. In the case of clustering, the number of items will reduce to $\binom{225}{5} = 4.5951\text{e+}009$. Hence, clustering of the elements leads to near 1800 times reduction in the size of search space. Thus, we see significant influence of clustering on reduction of computational effort for large-scale structures.

To ease explanation of the method, we define $V_i;\ i = 1,2,...nc$ as the vector subspaces spanned by the RSEs of the corresponding clusters, respectively. Mathematically, we can write V_i as:

$$V_i = \text{span}(\mathbf{S}_j : j \in C_i);\ \ i = 1,2,...nc \tag{5}$$

where C_i is the ith cluster and nc is the total number of clusters. The associated optimization problem can be written as:

Cost function: $$F(I) = \left\| \Delta\mathbf{U} - \text{proj}_{\sum_{i\in I} V_i} \Delta\mathbf{U} \right\| \tag{6}$$

Subject to: $$I \subset \{1,2,...nc\} \ \ \& \ \ |I| = l$$

in which the cost function, $F(I)$, should be minimized, where $\|.\|$ and $|.|$ denote Euclidean norm of a vector and the number of members of a set, respectively. l is the maximum number of damaged elements. Based on the standard definition of subspaces summation, $\sum_{i\in I} V_i$ is equal to $\text{span}(\mathbf{S}_i : i \in \bigcup_{j\in I} C_j)$. Noteworthy, l makes an upper bound for the number of damages in which the algorithm is valid. As an instance, in the case of setting l to 4, the proposed method is applicable only for the cases up to 4 damaged elements.

3.1.2. Discrete ACO for the first stage

ACO meta-heuristic was first proposed by Marco Dorigo [21] to tackle hard combinatorial optimization problems. The method mimics the social foraging behaviour of ants to identify the shortest path between food source and nest. As the ants move from food sources to the nest and vice versa, they leave a chemical liquid called pheromone on the ground, making a trail of pheromone. Ants can smell pheromone and they tend to choose the paths with more pheromone amount.

We employ ACO to solve the optimization problem of Eq. (6). To this aim, we consider l nodes in a circle pattern which are mutually connected by nc paths to each other as shown in Figure 2. Some artificial ants are randomly placed in the nodes of this hypothetical model. The ants are allowed to move in clockwise direction only. In each time step (iteration), each of the ants stochastically selects one of the paths from 1 to nc to go to its next node. During l time steps, the ants come into their initial nodes and thus their tours are completed. The selected paths for each ant specify a solution for the problem. That is, the selected paths denote the clusters. The procedure of making a tour by each of the ants is called a cycle. As cycles proceed, the solutions are improved toward the optimal ones (the solutions containing all damaged clusters). The number of ants (k) is constant during ACO.

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018

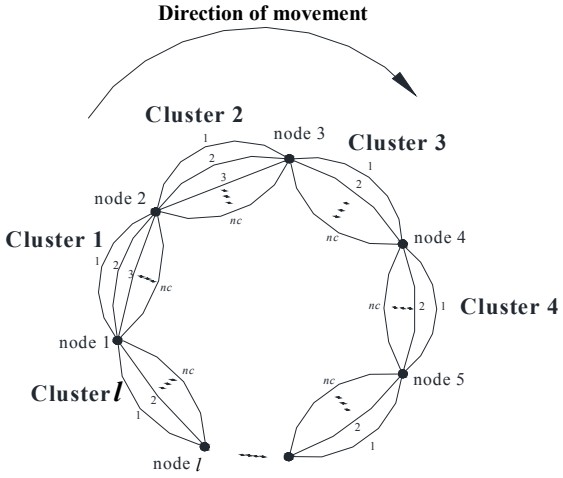


Figure 2. The model for selecting l clusters by ants

Here, we see how stochastically the ants select their paths. We specify each path by the ordered pair (i, j) indicating the path j that an ant locating at the node i moves along it. An ant selects the path (i, j) by the following probability:

$$p_{ij} = \frac{a_{ij}}{\sum_{q=1}^{nc} a_{iq}} \tag{7}$$

where a_{ij} is a value corresponding to path (i, j) which is evaluated using the relationship:

$$a_{ij} = \frac{\tau_{ij}^{\alpha}}{\sum_{q=1}^{nc} \tau_{iq}^{\alpha}} \tag{8}$$

where τ_{ij} is the pheromone intensity for path (i, j) at the current time, and α is a positive value by which the effect of pheromone is adjusted. Initially, a small pheromone value, say τ_0, is uniformly assigned to all paths, and thus all paths have equal chances to be selected by ants. Here, we employ ranked-based ant system (AS_{rank}) for global pheromone updating [22]. In this scheme, after each cycle, the solutions are ranked according to the fitness of objective function $F(I)$, from the best to worst. The pheromone value of the paths associated with high ranked ants and the elite ant is subsequently increased. The first λ solutions are considered as high ranked solutions, where λ is one of the parameters of ACO. The μth ranked solution and its corresponding cost functions are denoted by T^{μ} and F^{μ}, respectively. Also the tour and the cost function corresponding to the best-so-far solution are denoted by T^{+} and F^{+}. In AS_{rank}, the pheromone values concerning high ranked solutions and the best-so-far solution are increased using the following relationships:

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018

$$\Delta\tau_{ij}^{\mu}=\begin{cases}\dfrac{\lambda-\mu}{F^{\mu}} & , \text{ if } (i,j)\in T^{\mu} \ \& \ \mu\le\lambda \\ 0 & , \text{ otherwise}\end{cases} \tag{9-a}$$

$$\Delta\tau_{ij}^{+}=\begin{cases}\dfrac{\lambda}{F^{+}} & , \text{ if } (i,j)\in T^{+} \\ 0 & , \text{ otherwise}\end{cases} \tag{9-b}$$

where $\Delta\tau_{ij}^{+}$ and $\Delta\tau_{ij}^{\mu}$ are increments in pheromone value for path (i,j) which concern the best solution and the μ th ranked solution, respectively.

Evaporation of phenomenon helps real ants to escape from maturing. Evaporation is simulated in ACO by reducing the amount of pheromone deposited from past cycles. The evaporation procedure enables ACO to explore new areas in the search space and to escape from local minimums. The pheromone value of path (i,j) at the next cycle in the case of considering evaporation is given by the following formula:

$$\tau_{ij}(t+l)\leftarrow\rho.\tau_{ij}(t)+\Delta\tau_{ij}^{+}(t)+\sum_{\mu=1}^{\lambda\text{-}1}\Delta\tau_{ij}^{\mu}\ (t) \tag{10}$$

where $\rho\in(0,1]$ is the evaporation rate.

This stage consists of seven steps as follows:

1. Establish the sensitivity matrix **S** by Eq. (4), and evaluate VRC.
2. Partition the elements into same size clusters randomly. If the number of elements is not dividable into the number of clusters, duplicate some of the elements to make it dividable.
3. Consider the hypothetical model of Figure 2 and assign initial pheromone values to the paths uniformly.
4. Put the ants randomly in the joints of the model and select l clusters via ants considering the probabilities of Eq. (7).
5. After all ants finish their tour, evaluate the objective function $F(I)$.
6. Identify the elite and best rank ants. Update their path trace using Eq. (10).
7. If the convergence criterion is met, get the best-so-far solution as the final result; otherwise, go to Step 4.

3.2. The second stage

3.2.1. The objective function for the second stage

In the second stage, the elements constituted the extracted clusters from the first stage are considered as damage candidates. These reduced elements are referred to restricted elements (RE) and denoted by the set $RE=\{re_1,re_2,\ldots,re_{\bar{n}}\}\subset\{1,2,\ldots,n\}$, where

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018

re_i; $i = 1,2,...\bar{n}$ are the restricted elements and $\bar{n}$ is the number of restricted elements. Note that when we use RE in the italic form, *RE*, we refer to the restricted elements as a set. If the *i*th element is a member of *RE*, this element is considered as damage candidates and, therefore, $x_i \in [0,1]$. On the flip side, if the *i*th element is not contained in *RE*, that element is considered healthy, and thus $x_i = 0$.

In this stage, the damage extents of the restricted elements (RE) are determined using the continuous version of ACO [7]. Due to the reduction in the number of design variables at the first stage, Eq. (3) becomes over-determined. Hence, by best equalizing the sides of Eq. (3), the true solution is obtained. Thus, the objective function $D = \left\| \mathbf{U}_d - \mathbf{U}(\bar{\mathbf{X}}) \right\|$ should be minimized, where $\bar{\mathbf{X}}$ is the reduced damage vector with the entries associated with RE, i.e. $\bar{\mathbf{X}} = (x_{re_1}, x_{re_2}, ...x_{re_{\bar{n}}})$. It should be noted that all elements of RE are not essentially damaged, and there may exist some healthy ones among them, i.e. $SDE \subseteq RE$.

3.2.2. Continuous ACO for the second stage

In continuous ACO, the ants sample each of the components of the solutions from a continuous probability density function (PDF) in the feasible domain. The PDFs gradually changes in a way to bias the sampling toward the optimum solution during implementation of ACO. This kind of ACO is denoted by ACO_R [23]. The best solutions in each cycle are kept for the next cycle and the other ones are replaced with the newly produced solutions. The Gaussian kernel PDFs are adapted for the associated PDFs. These PDFs are constructed using weighted sum of several one dimensional Gaussian functions. Gaussian kernel PDFs are capable of producing PDFs with several rises. We write the associated Gaussian kernel for x_{re_i}; $i = 1,2,...\bar{n}$ as:

$$G_i(x) = \sum_{z=1}^{k} \omega_z g_i^z(x) = \sum_{z=1}^{k} \omega_z \frac{1}{\sigma_i^z \sqrt{2\pi}} \exp\left(\frac{-\left(x - \mu_i^z\right)^2}{2\left(\sigma_i^z\right)^2} \right) \tag{11}$$

where $g_i^z(x)$ is a Gaussian PDF having mean μ_i^l and standard deviation σ_i^l. In ACO_R, we also denote the number of ants by *k*. In each cycle, the *k* ants make *k* solutions as $\bar{\mathbf{X}}^i = (x^i_{re_1}, x^i_{re_2}, ...x^i_{re_{\bar{n}}})$; $i = 1,2,...k$ in which $x^i_{re_j}$ denotes the *j*th variable of the *i* th reduced solution. The objective function for all *k* solutions is computed. The solutions are ordered from the best to worst accordingly. Afterwards, the solutions are stored in an archive as shown in Figure 3.

The weights of the solutions are evaluated as:

$$\omega_z = \frac{1}{qk\sqrt{2\pi}} \exp\left(\frac{-(z-1)^2}{2q^2k^2} \right) \tag{12}$$

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018

where z is the rank of each solution and q is a parameter of ACO_R which should be assigned at first. Subsequently, the determined ω_z's are used in Eq. (11). Notice that, $\omega_k \le \cdots \le \omega_2 \le \omega_1$.

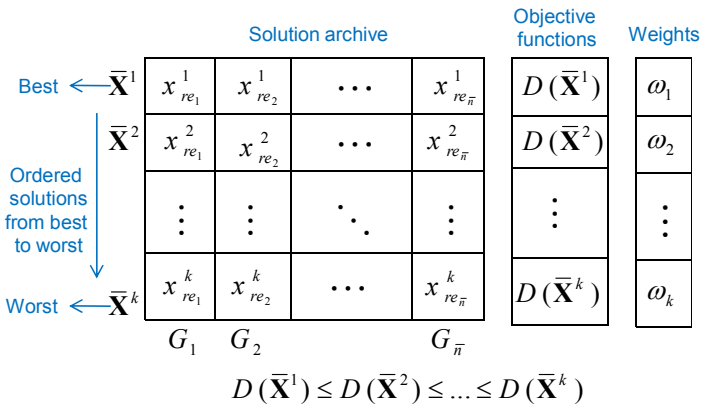


Figure 3. The solution archive used in ACO_R

As expressed before, for sampling of solutions of the next cycle by ants, the PDFs $G_i(x);\ i=1,2,...\bar{n}$ are needed. Hence, the values of μ_i^z and σ_i^z for $z=1,2,...k$ and $i=1,2,...\bar{n}$ should be first assigned. The values of μ_i^z $(z=1,2,...k, i=1,2,...\bar{n})$ are respectively considered as $x^z_{re_i}$ $(z=1,2,...k, i=1,2,...\bar{n})$ which are the components of the current solutions. The standard deviations are assigned as $\sigma_i^z = \xi \sum_{e=1}^{k} \frac{\left| x^e_{re_i} - x^z_{re_i} \right|}{k-1}$ where ξ is a constant playing the role of evaporation rate. To make the solutions in the first cycle, we employed the PDFs $G_i(x) = \sum_{z=1}^{k} \frac{1}{k} g_i^z\left(x, (2z-1)\frac{1}{2k}, \frac{1}{2k}\right)$ [24].

The step by step summary of the second stage is as follows:

1. Construct initial solutions by ants.
2. Calculate the objective function for the current solutions, $\bar{\mathbf{X}}^i;\ i=1,2,...k$.
3. Construct $G_i(x);\ i=1,2,...\bar{n}$ using the current solution archive.
4. Keep the best solutions of the current cycle for the next cycle and omit the others.
5. Construct new solutions to replace the omitted solutions. Sample the *i*th component of the new solutions using $G_i(x);\ i=1,2,...\bar{n}$.
6. Iterate Steps 2 to 5, until the convergence criteria are met.

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018

4. CASE STUDIES

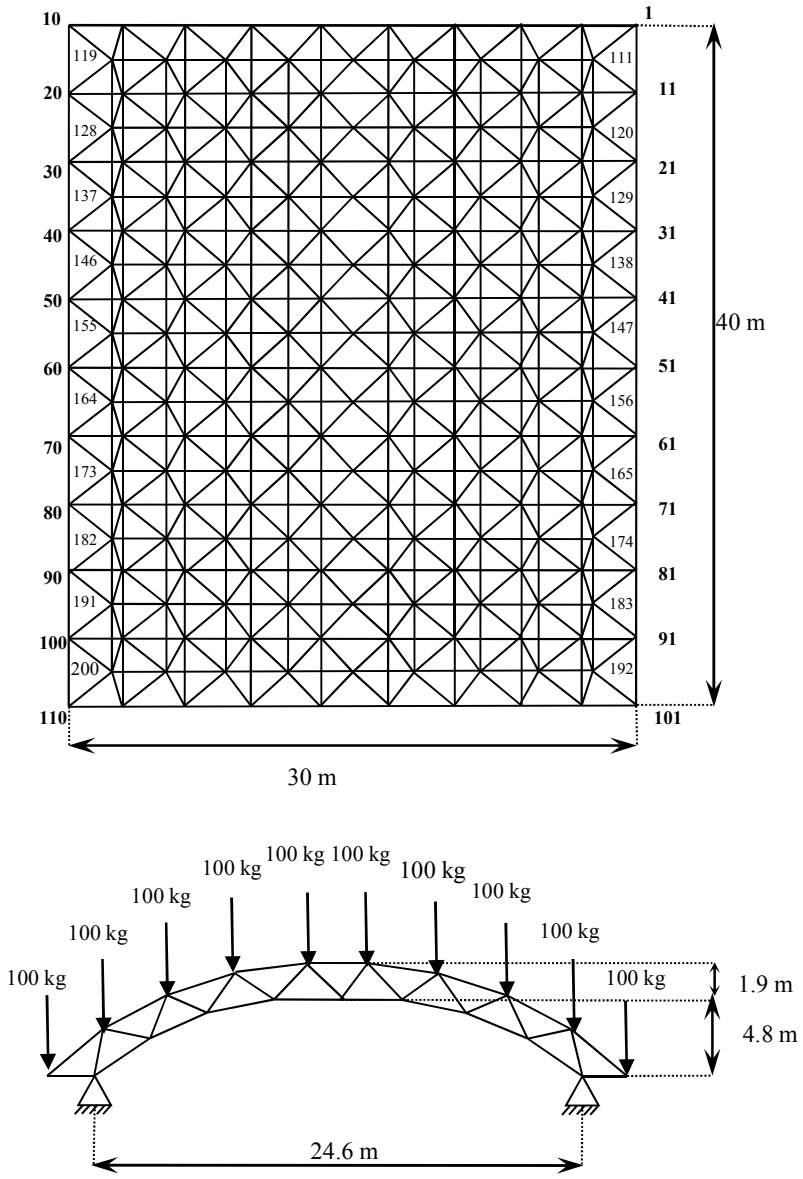
In this section, the efficiency of the proposed method is illustrated by two large-scale structures. Young's modulus for steel materials in both case studies is considered as 2.1×10^6 kg/cm^2. To identify the damaged elements, we measure the static deflections in gravity direction for all joints of the structures. In these two examples, the damage is simulated by a reduction in the Young's modulus of the damaged elements with the amount of damage severity.

4.1. The first case study

A 721-element barrel vault is considered with the dimensions shown in Figure 4a. In this figure the joint numbers are given. The structure is supported at the corner joints of bottom layer as shown. Concentrated loads with magnitude of 100 kg are applied to all joints of top layer in the gravity direction. Cross sectional areas of all elements are 40cm^2. Damage detection of the structure under two damage scenarios is considered. These two scenarios are shown in Figure 4b and c in which the damaged elements are highlighted. In the figure, the damage element numbers are illustrated. Also, the percentages of damages are given in the parentheses.

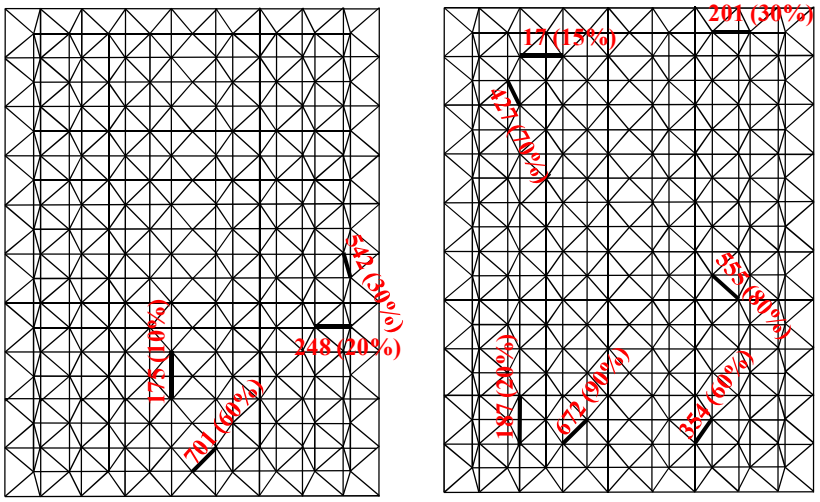
As aforementioned, the first stage of the proposed algorithm reduces the damage variables. The parameters of discrete ACO in the first stage of the algorithm for Scenarios 1 and 2 are given in Table 1a and b, respectively. For this stage the cluster size is selected to be four. Therefore, the number of design variables is equal to 181 (the ratio of 721 to 4 is equal to 180.25). After the first stage, the suspended elements are reduced to 40 and 60 for Scenarios 1 and 2, respectively. However, for more reduction in damage variables, the first stage in executed once more. After the second run of the first stage, the design variables are reduced to 10 for both scenarios. These ten elements are listed in Table 2 in which the true damaged ones are bold. Figure 5 illustrates the convergence history of *F(I)* for both average and the elite solutions in the first run and second run of ACO at the first stage. As it can be seen, in all cases the objective function truly converges to a very small value near zero. Noteworthy, clustering of elements is highly effective in performance of the first stage. That is, in the case of no clustering the first stage works much weaker. Based on our numerical results, in such a case, the first stage of algorithm is unable to identify all damaged elements. Namely, some of the damaged elements are missed from the reduced design variables. This event is due to the fact that in the case of no clustering, the search space is very large and has many local minimums. Hence, ACO enables to find the true solution among them. The parameters of ACO_R in the second stage are presented in Table 3. Figure 6 illustrates the convergence history of the objective function $D = \left\| \mathbf{U}_d - \mathbf{U}(\bar{\mathbf{X}}) \right\|$ for the second stage and both scenarios. The objective function converges to zero. This indicates that the damage extents are correctly obtained through the ACO_R process. Figure 7 yields convergence history of the damage severities for the ten remained elements during the ACO_R procedure. For clarity, in this figure convergence histories of damaged and healthy elements are given separately. The obtained damage extents for the intact elements are near zero which is desirable.

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018



(a)

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018



(b) (c)

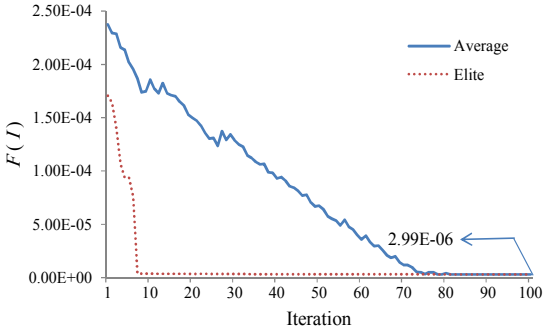
Figure 4. (a) Geometry of the 721-element barrel vault, (b) Scenario 1, and (c) Scenario 2

Table 1a. Parameters of discrete ACO for the first stage in Scenario 1

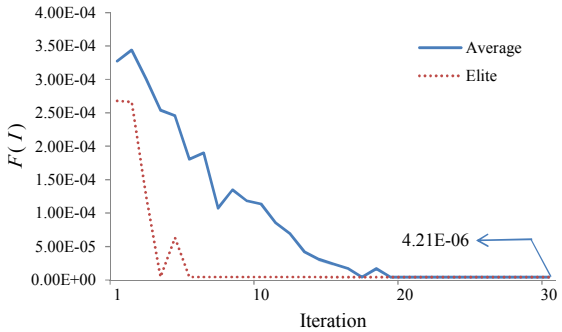
Run	α	β	k	λ	ρ_1	ρ_2	τ_0	n	l
1	0.2	0	100	10	0.5	0.5	1000	720/4	10
2	0.2	0	20	2	0.5	0.5	1000	40	10

Table 1b. Parameters of discrete ACO for the first stage in Scenario 2

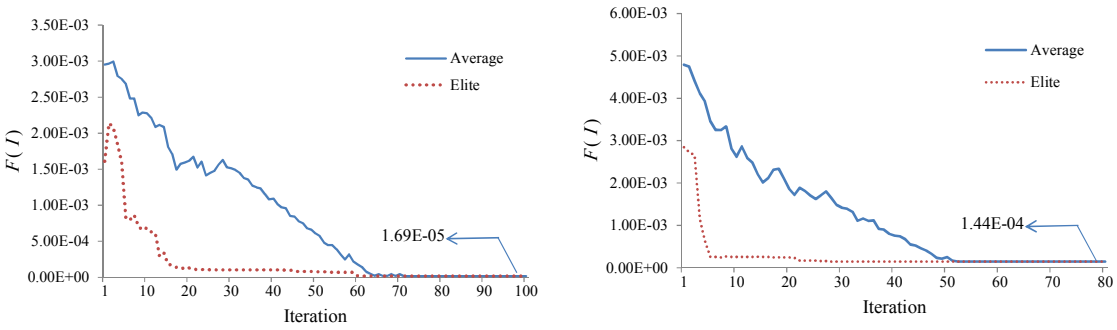
Run	α	β	k	λ	ρ_1	ρ_2	τ_0	n	l
1	0.2	0	70	7	0.5	0.5	1000	181	15
2	0.2	0	30	3	0.5	0.5	1000	60	10



(a)



(b)



(c) (d)

Figure 5. Convergence histories of objective function during run of the first stage: (a) first run for Scenario 1, (b) second run for Scenario 1, (c) first run for Scenario 2, and (d) second run for Scenario 2

Table 2a. Remained design variables after twice execution of the first stage of algorithm for Scenario 1

Label of elements	Label of elements by joints	Label of elements	Label of elements by joints
175	76-86	542	52-156
184	85- 95	571	68-163
218	131-132	629	75-178
248	165-166	701	95-196
541	51- 156	703	105-196

Table 2b. Remained design variables after twice execution of the first stage of algorithm for Scenario 2

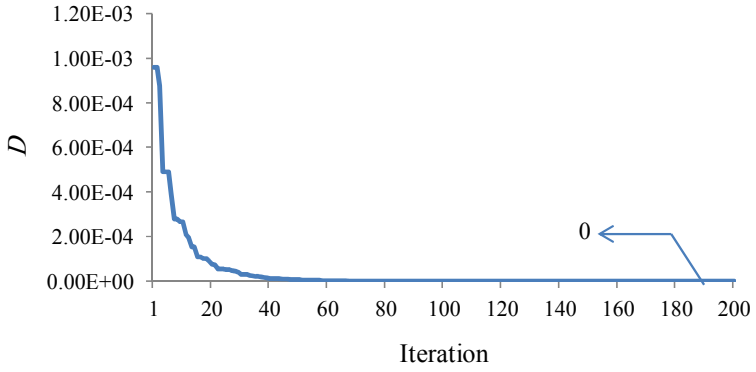
Label of elements	Label of elements by joints	Label of elements	Label of elements by joints
17	18-19	430	20-128
187	88-98	472	42-138
201	112-113	555	64-159
354	185-194	668	96-187
427	28-127	672	97-188

Table 3. Parameters of ACO_R for both scenarios

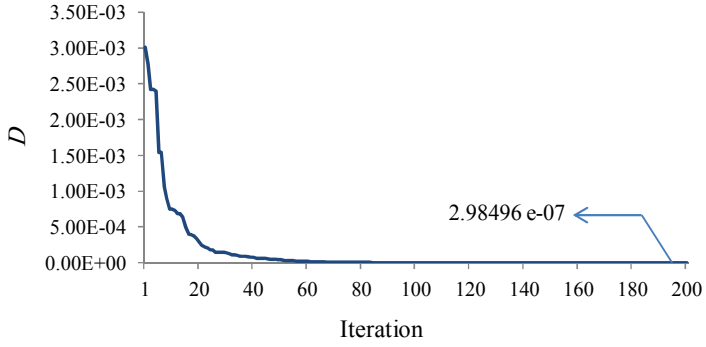
m	ξ	q	k
20	1.3	0.2	30

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018

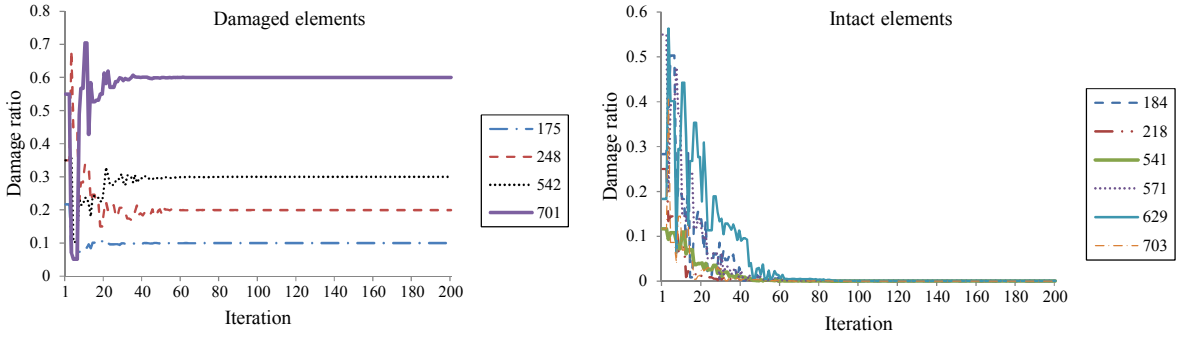


(a)

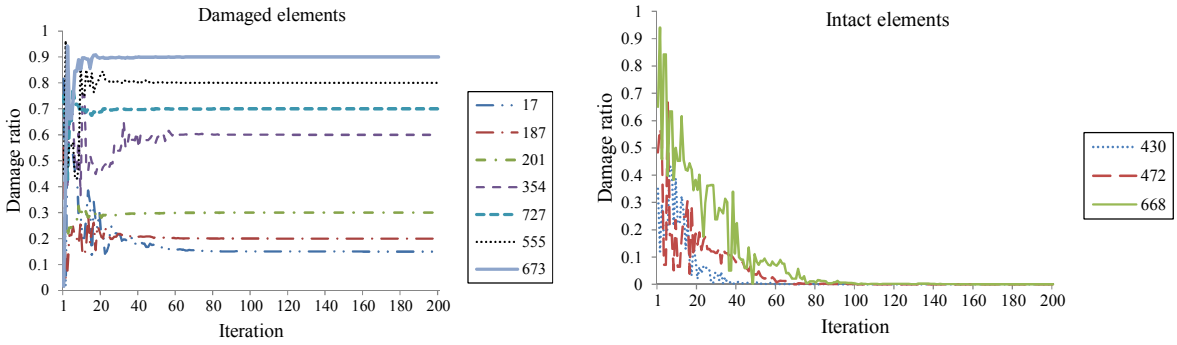


(b)

Figure 6. Convergence histories of objective function for the second stage: (a) for Scenario 1, and (b) for Scenario 2



(a)

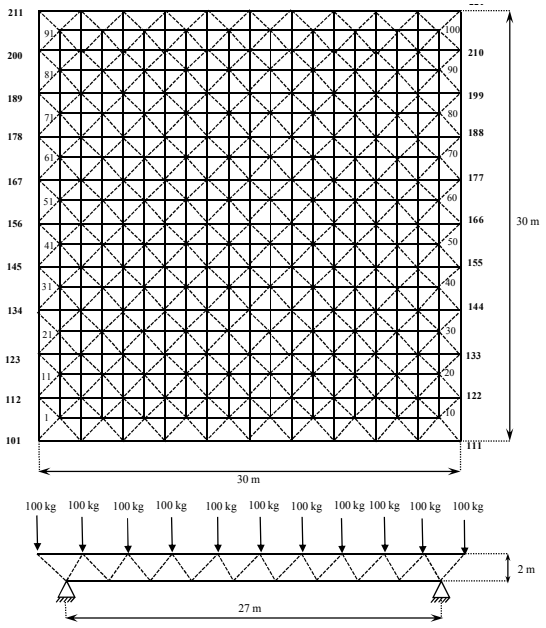


(b)

Figure 7. Convergence history of ten suspended elements in second stage: (a) Scenario 1, and (b) Scenario 2

4.2. The second case study

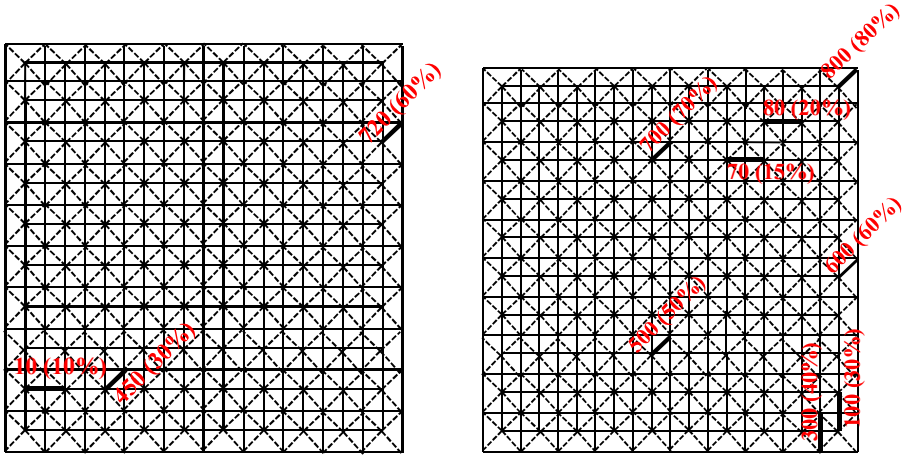
A double-layer grid with 800 elements is considered as shown in Figure 8a. The length, width, and height of the structure are 30 m, 30 m and 2 m, respectively. The structure is supported at the joints 1, 10, 91, and 100. Concentrated loads with magnitude of 100 kg are applied to joints of the top grid in the gravity direction. The cross sectional area for all elements is 40 cm^2. Two damage scenarios with three and eight damaged elements are to be considered. Figure 8b and c shows the pattern of these two scenarios with damaged element numbers and amounts of damage.



(a)

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018



(b) (c)

Figure 8. (a) A 800-element double-layer grid, (b) Scenario 1, and (c) Scenario 2

Like the previous case study, the first stage is executed twice for more reduction in design variables. The cluster sizes are considered four. Table 4 gives the parameters utilized in ACO for the first stage.

Table 4a. Parameters of discrete ACO for the first stage and Scenario 1

Run	**α**	**β**	**K**	**λ**	**ρ_1**	**ρ_2**	**τ_0**	**n**	**l**
1	0.2	0	100	10	0.5	0.5	1000	400	10
2	0.2	0	10	1	0.5	0.5	1000	20	10

Table 4b. Parameters of discrete ACO for the first stage and Scenario 2

Run	α	β	K	λ	ρ_1	ρ_2	τ_0	n	l
1	0.2	0	80	8	0.5	0.5	1000	200	10
2	0.2	0	30	3	0.5	0.5	1000	40	10

Figure 9 depicts the convergence history of the objective function $F(I)$ for the first and second run of the first stage for both scenarios. After the second run of the first stage, only 10 suspected elements are remained as design variables. Table 5 represents ten remained

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018

elements from the first stage for both scenarios. In this table, the true damaged elements are bold. As seen, the remained elements truly consist of all damaged elements. Table 6 represents the parameters of ACO_R for the second stage of the algorithm. Figure 10 shows convergence histories of the objective function D which correctly converges to zero for both scenarios. Finally, Figure 11 depicts convergence history of the design variables during implementation of ACO_R in the second stage. As it can be seen, the damage severities of healthy elements of RE are accurately obtained as zero.

Table 5a. Remained design variables after twice execution of the first stage of algorithm for Scenario 1

Label of elements	Label of elements by joints	Label of elements	Label of elements by joints
10	11-12	450	13-115
107	17-27	570	43-148
134	44-54	620	55-172
314	124-135	663	66-183
379	189- 200	720	80-199

Table 5b. Remained design variables after twice execution of the first stage of algorithm for Scenario 2

Label of elements	Label of elements by joints	Label of elements	Label of elements by joints
70	77-78	300	110-121
80	88- 89	500	25-139
100	10-20	600	50-166
117	27-37	700	75-194
268	196-197	800	100-221

Table 6. Parameters of ACO_R for both scenarios

m	ξ	q	k
20	1.3	0.2	30

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018

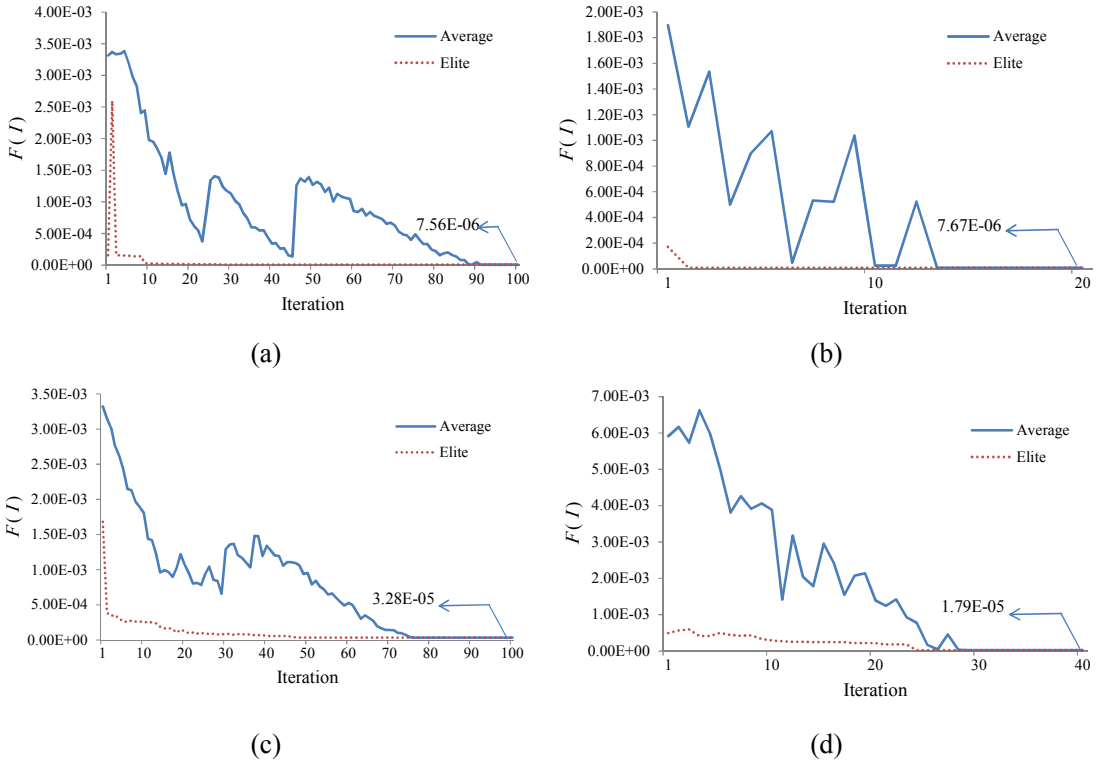


Figure 9. Convergence histories of objective function during run of the first stage: (a) first run for Scenario 1, (b) second run for Scenario 1, (c) first run for Scenario 2, and (d) second run for Scenario 2

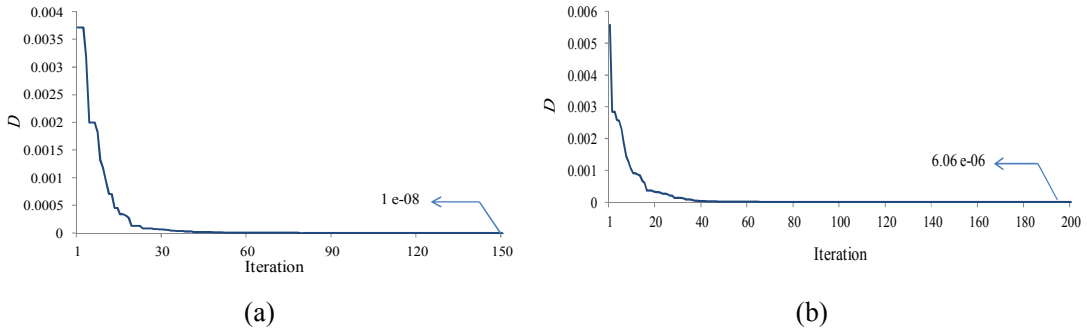
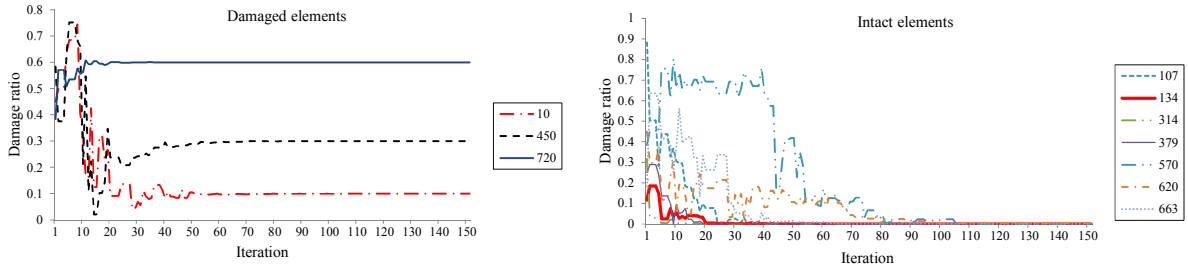
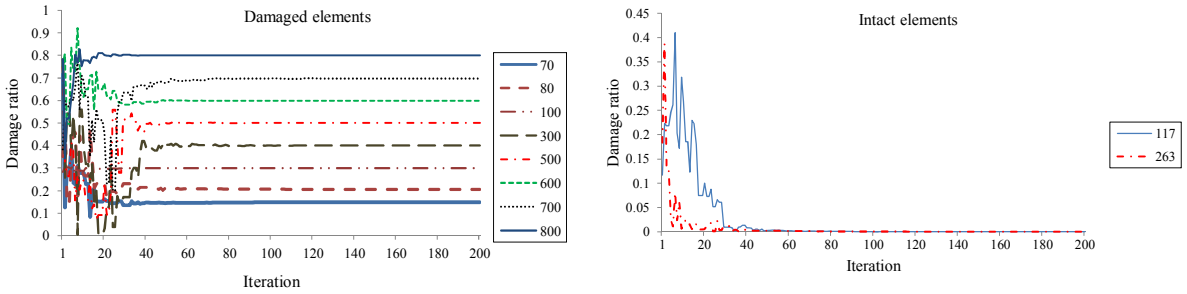


Figure 10. Convergence histories of objective function for the second stage: (a) for Scenario 1, and (b) for Scenario 2

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018



(a)



(b)

Figure 11. Convergence history of ten suspended elements in the second stage: (a) Scenario 1, and (b) Scenario 2

5. CONCLUSIONS

In this work, an efficient two-stage technique is presented for damage detection of large-scale structures. In the first stage of method, the design variables are reduced by employing discrete version of ACO and utilizing sensitivity matrix of structure. In the second stage, a continuous version of ACO detects the damage extents of the restricted elements by minimizing the difference of static responses between hypothetically damaged and real damaged structures. The performance of the proposed procedure is investigated by a 721-element and an 800-element space structure. For both cases, all damaged elements exist in the reduced damage candidates of the first stage. This is also true when the first stage is executed twice for more reduction in damage candidates. In the second stage, ACO_R correctly identifies the damage extents of the reduced damage variables. Noteworthy, in both cases the damage extents of the healthy elements are truly assigned to be zero. Finally, it is concluded that the proposed method is capable of identifying the damages in the large-scale structures accurately with a low computational burden.

REFERENCES

1. Koh CG, Chen YF, Liaw CY. A hybrid computational strategy for identification of

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018

structural parameters, *Comput Struct,* 2003; **81**:107–17.
2. Koh BH, Dyke SJ. Structural health monitoring for flexible bridge structures using correlation and sensitivity of modal data, *Comput Struct,* 2007; **85**:117–30.
3. Vakil-Baghmisheh MT, Peimani M, Homayoun Sadeghi M, Ettefagh MM. Crack detection in beam-like structures using genetic algorithms, *Appl Soft Comput,* 2008; **8**:1150–60.
4. Begambre O, Laier JE. A hybrid particle swarm optimization - simplex algorithm (PSOS) for structural damage identification, *Adv Eng Softw,* 2009; **40**:883–91.
5. Nasealavi SS, Salajegheh J, Salajegheh E, Fadaee MJ. An improved genetic algorithm using sensitivity analysis and micro-search for damage detection, *Asian J Civil Eng,* 2010; **11**(6):717–40.
6. Meruane V, Heylen W. An hybrid real genetic algorithm to detect structural damage using modal properties, *Mech Syst Signal Pr,* 2011; **25**:1559–73.
7. Yu L, Xu P. Structural health monitoring based on continuous ACO method. *Micro Reliab* 2011; **51**:270–8.
8. Au FTK, Cheng YS, Tham LG, Bai ZZ. Structural damage detection based on a Microgenetic algorithm using incomplete and noisy test data, *J Sound Vib,* 2003; **259**(5):1081–94.
9. Friswell MI, Penny JET, Garvey SD. A combined genetic and eigensensitivity algorithm for the location of damage in structures, *Comput Struct,* 1998; **69**:547–56.
10. Guo HY, Li ZL. A two-stage method to identify structural damage sites and extents by using evidence theory and micro-search genetic algorithm, *Mech Syst Signal Pr,* 2009; **23**:769–82.
11. Naseralavi SS, Fadaee MJ, Salajegheh J. Subset solving algorithm: a novel sensitivity-based method for damage detection of structures, *Appl Math Mode,* 2011;35:2232–52.
12. He R, Hwang S. Damage detection by a hybrid real-parameter genetic algorithm under the assistance of grey relation analysis, *Eng Appl Artif Intell,* 2007; **20**:980–92.
13. Jenkins CH, Kjerengtroen L, Oestensen H. Sensitivity of parameter changes in structural damage detection, *Shock Vib,* 1997; **10**(4),135–42.
14. Cao M, Ye L, Zhou L, Su Z, Bai R. Sensitivity of fundamental mode shape and static deflection for damage identification in cantilever beams, *Mech Syst Signal Pr,* 2011; **25**:630–43.
15. Wang X, Hu N, Fukunaga H, Yao ZH. Structural damage identification using static test data and changes in frequencies, *Eng Struct,* 2001; **23**:610–21.
16. Banan MR, Banan MR, Hjelmstad KD. Parameter estimation of structures from static response, I: computational aspects, *J Struct Eng,* 1994a; **120**(11):3243–58.
17. Banan MR, Banan MR, Hjelmstad KD. Parameter estimation of structures from static response, II: numerical simulation studies. *J Struct Eng,* 1994b; **120**(11):3259–83.
18. Bakhtiari-Nejad F, Rahai A, Esfandiari A. A structural damage detection method using static noisy data, *Eng Struct,* 2005; **27**:1784–93.
19. Wu JR, Li QS. Structural parameter identification and damage detection for a steel structure using a two-stage finite element model updating method, *J Construct Steel Res,* 2006; **62**:231–9.
20. Titurus B, Friswell MI, Starek L. Damage detection using generic elements: Part II,

damage detection, *Comput Struct,* 2003; **81**:2287–99.
21. Dorigo M, Caro DG, Gambardella LM. Ant algorithms for discrete optimization, *Artif Life,* 1999; **5**(3):137–72.
22. Bullnheimer B, Hartl RF, Strauss C. *A New Rank-based Version of the Ant System*, a computational study, Technical Report POM-03/97, Institute of Management Science, University of Vienna, 1997.
23. Sosha K, Dorigo M. Ant colony optimization for continuous domains, *Eur J Oper Res,* 2008; **185**:1155–73.
24. Madadgar S, Afshar A. An improved continuous algorithm for optimization for water resources Problems, *Water Resour Manage,* 2009; **23**(10):2119–39.

Downloaded from www.iust.ac.ir at 3:52 IRDT on Friday March 23rd 2018