Smart Structures and Systems, *Vol. 18, No. 5 (2016) 983-1004*
DOI: http://dx.doi.org/10.12989/sss.2016.18.5.983

Detection of damage in truss structures using Simplified Dolphin Echolocation algorithm based on modal data

Ali Kaveh[*1], Seyed Rohollah Hoseini Vaez[2], Pedram Hosseini[2] and Narges Fallah[2]

[1]*Centre of Excellence for Fundamental Studies in Structural Engineering, School of Civil Engineering, Iran University of Science and Technology, Tehran-16, Iran*
[2]*Department of Civil Engineering, Faculty of Engineering, University of Qom, Iran*

(Received January 17, 2016, Revised June 3, 2016, Accepted June 10, 2016)

Abstract. Nowadays, there are two classes of methods for damage detection in structures consisting of static and dynamic. The dynamic methods are based on studying the changes in structure's dynamic characteristics. The theoretical basis of this method is that damage causes changes in dynamic characteristics of structures. The dynamic methods are divided into two categories: signal based and modal based. The modal based methods utilize the modal properties consisting of natural frequencies, modal damping and mode shapes. As the modal properties are sensitive to changes in the structure, these can be used for detecting the damages. In this study, using dynamic method and modal based approach (natural frequencies and mode shapes), the objective function is formulated. Then, detection of damages of truss structures is addressed by using Simplified Dolphin Echolocation algorithm and solving inverse optimization problem. Many scenarios are used to simulate the damages. To demonstrate the ability of the algorithm, different truss structures with several multiple elements scenarios are tested using a few runs. The influence of the two different levels of noise in the modal data for these scenarios is also considered. The last example of this article is investigated using a different mutation. This mutation obtains the exact answer with fewer loops and population by limited computational effort.

Keywords: damage detection; SDE algorithm; truss structures; frequencies; mode shapes; inverse problem

1. Introduction

Structural health monitoring is a process for getting exact information about structure's conditions and performance instantaneously. Detecting the structure's unusual behaviors is the main purpose of monitoring which shows undesirable conditions. The data obtained from monitoring are used for optimizing the performance, maintenance, repair and/or replacing the structural components based on reliable and measured data. In the monitoring topic, damage means some changes occurring within utilizing the structure; and identifying damage includes all the techniques and methods used for detecting the damages and their locations and severity (Doebling, Farrar *et al*. 1996, Sohn *et al*. 2004).

Considering the changes that happen in the modal parameters after damage is one of the efficient means for detecting damage. There is a complete review of damage detection

∗Corresponding author, Professor, E-mail: alikaveh@iust.ac.ir

http://www.techno-press.org/?journal=sss&subpage=8 ISSN: 1738-1584 (Print), 1738-1991 (Online)

methodologies based on dynamic parameters in (Doebling, Farrar *et al*. 1998, Fan and Qiao 2004, Carden and Fanning 2004). Villalba and Laier (2012) used a methodology for detecting and quantifying damages in trusses by using multi-chromosome genetic algorithm based on changes in the natural frequencies and mode shapes. Other damage detection methods can be found in the work of Hakim and Abdul Razak (2013), Kaveh and Maniat (2015), Pan, Yu *et al*. (2016).

Kaveh and Zolghadr (2015) employed two different objective functions to acquire all the global optimal solutions. The first objective function is only based on natural frequencies therefore, if the tested structure with this objective function is symmetric, there will be no unique result. They used this objective function intentionally to demonstrate the ability of the provided algorithm in finding multiple global optimal solutions in few runs. The second objective function which was based on natural frequencies and mode shapes was utilized for detecting the exact locations and severity of the damages.

For detecting steel trusses damage, Kaveh and Mahdavi (2016) applied Colliding Bodies Optimization (CBO) and Enhanced Colliding Bodies Optimization (ECBO) algorithms, Kaveh (2014). Also they employed the dynamic parameters of undamaged structure to formulate the objective function in addition to the parameters of the damaged structure.

Majumdar *et al*. (2012) used ant colony algorithm to detect truss structures damage based on changes in the natural frequencies. They obtained the frequencies of undamaged structure by utilizing their own method and the method introduced by Kwon and Bang (2000). Then they gained reliable results for detecting damages by comparing these parameters with parameters obtained from the ant colony algorithm for the damaged structure. Kaveh and Maniat (2014) applied the CSS algorithm to detect damage in a continuous beam, a three-story and three-span plane frame, a planar truss and a spatial truss using incomplete data.

One of the recently developed metaheuristics is Dolphin Echolocation (DE) introduced by Kaveh and Farhoudi (2013, 2016). This algorithm has been simplified and modified by Kaveh and Hosseini (2014) to introduce the Simplified Dolphin Echolocation (SDE) method.

The DE algorithm and its simplified version (SDE) are based on dolphin's hunting technique. Dolphins send the voice in different directions and listen to its echo and thereby find the location of their baits and move towards them. While approaching the baits, dolphins send waves continually and thus the probability of successful hunting increases more and more. The optimal solution acts as model's bait in the algorithm. A significant change applied to the SDE algorithm, includes the decision making after getting information. As the algorithm progresses further, the decision making becomes more conceivable. However, the rate and accuracy of reaching to baits increases.

In this study SDE algorithm is used for detecting damages of truss structures using the information on changes in natural frequencies and mode shapes. The remaining of this paper is organized as follows: the formulation of the problem is provided in section 2. In section 3, the optimization algorithm consisting of the SDE algorithm and a brief overview of the standard DE are presented. In section 4, numerical examples are examined. Finally, discussions and concluding remarks are presented in section 5.

2. Formulation of problem

2.1 Finite element model for undamaged structure

For preparing finite element model, first the stiffness and mass matrices are calculated for each element. Then stiffness and mass matrices of the structure for undamaged state, denoted by *Ks* and *Ms*, respectively, are constructed as follows

$$K_s = \sum_{j=1}^{Ne} k_j \tag{1}$$

$$M_s = \sum_{j=1}^{Ne} m_j \tag{2}$$

where k_j and m_j are the stiffness and mass matrices of the jth element, respectively; *Ne* is the number of elements of the structure.

2.2 Evaluating the dynamic parameters of the undamaged structure

The undamaged dynamic parameters are computed from the following eigenvalue equation

$$([K_s] - \omega_{ju}^2 [M_s])\{\varphi_{ju}\} = \{0\} \tag{3}$$

where ω_{ju} and φ_{ju} are the *j*th natural frequency and mode shape of the undamaged state, respectively. It is worth mentioning that the analyzed structure is undamped.

2.3 Simulating the damage by reduction of modulus of elasticity

Damage detection is an inverse optimization problem and applies a set of parameters corresponding to damaged elements in the finite element model, to simulate the damaged structure. Damage is identified by reduction of the structural properties. In this study, the damage is identified using reduction in element's modulus of elasticity. Therefore, the relationship between the two states for the *j*th element is obtained by

$$E_{jd} = (1 - \beta_j) \times E_j \tag{4}$$

in which, E_{jd} and E_j are the moduli of elasticity of the *j*th element for the damaged and undamaged states, respectively; β_j is the modulus of elasticity reduction coefficient being 0 if the element is undamaged, and 1 for the completely damaged state. The value of β_j varies in different scenarios. The reduction in modulus of elasticity causes reduction of stiffness matrix as following

$$[K_d] = \sum_{j=1}^{Ne} (1 - \beta_j)[k_j] \tag{5}$$

2.4 Evaluating the dynamic parameters of the damaged structure

where K_d is the stiffness matrix of structure for damaged state. Finally, the damaged dynamic parameters are obtained by

$$([K_d] - \omega_{jd}^2 [M_s])\{\varphi_{jd}\} = \{0\} \tag{6}$$

where ω_{jd} and φ_{jd} are the jth natural frequency and mode shape of the damaged state.

2.5 Adding noise to experimental dynamic parameters

Avoiding the noise is impossible in real dynamic tests, and therefore this issue is dealt with generating small deviation in experimental dynamic parameter as follows

$$\omega_{noise} = \omega_{jd} \times (1 + random\left(\text{-}1,1\right) \times Noise_f) \tag{7}$$

$$\varphi_{ijnoise} = \varphi_{i,jd} \times (1 + random\left(\text{-}1,1\right) \times Noise_\varphi) \tag{8}$$

where *noise* implies a noisy value; $Noise_f$ and $Noise_\varphi$ are the deviations of the natural frequencies and mode shapes, respectively (Villalba and Laier 2012, Chen and Nagarajaiah 2013)..

2.6 Formulating the objective function

In this section, minimization problem is formulated using an objective function which is based on natural frequencies and mode shapes as follows

$$F = \sum_{j=1}^{n}\left(\left|\frac{\omega_j^{SDE} - \omega_j^m}{\omega_j^m}\right| + \left|\sum_{i=1}^{ndof}\left(\frac{\left|\varphi_{ij}^{SDE}\right| - \left|\varphi_{ij}^m\right|}{\left|\varphi_{ij}^m\right|}\right)\right|\right) \tag{9}$$

in which, n is the number of the considered vibration modes, and $ndof$ is the number of degrees of freedom involved in the objective function; ω_j is the jth natural frequency and $|\varphi_j|$ is the absolute value of the jth mode shape for the ith degree of freedom. The superscript *SDE* and m refers to results from the finite element model computed by the *SDE* algorithm and the measured (experimental) values, respectively.

3. Optimization algorithm

3.1 Simplified Dolphin Echolocation

In this section, the SDE algorithm and its difference with the standard DE algorithm are briefly presented. Like many other metaheuristic algorithms, two stages of the SDE are the exploration and exploitation stages. In exploration and exploitation stages, the algorithm explores all the search space to perform a global search and then focuses on limited domain to search for better answers, respectively. When hunting process is started by dolphin, the probability of successful hunting increases every instant. PLi is the probability of the ith loop (In the remaining sections of this paper, loop means iteration), and determines the two stages of the search and is calculated by

$$P_{Li} = P_{L1} + (1 - P_{L1})\frac{Li - 1}{LN - 1} \tag{10}$$

where Li and LN are the number of the ith loop and number of loops, respectively; P_{L1} is the probability of the first loop which is usually obtained approximately as 0.1 (10%). This has not high impact on the convergence and accuracy of the final results, and can be calculated as follows

$$P_{L1} = \frac{\sum_{j=1}^{N} C_i}{N} \tag{11}$$

$$C_i = \frac{Mode_i}{M} \tag{12}$$

where N and M are the number of variables and number of locations, respectively; Ci is the modal among all answers; $Mode_i$ is the parametric mode having max iteration for each variable. When P_{Li} value is low, the algorithm is in exploration stage but when this parameter increases, the search space converts gradually to exploitation stage; and this parameter increases in each loop so that it meets one (100%) in the last loop. The search space becomes smaller continuously in every loop, and the algorithm changes to local search stage. Using Eq. (11) indicates that the mean value of P_{L1} is approximately equal to 10%, and in practice it can be observed that the use of this approximate value in place of the exact value does not have a significant effect on the final result.

The DE algorithm has a power parameter. This parameter specifies the increment of PLi to be linear or nonlinear.

When the value of power equals to one, the desirable accuracy is gained, i.e., the algorithm uses a linear function for changing exploration stage to exploitation stage. Power parameter being equal to one is equivalent to using a linear increment for the $PL1$. AC parameter is a new parameter introduced for showing the accuracy of every variable in the SDE algorithm; and it displays the number of decimal places for each variable. Moreover, the DE algorithm has another parameter named R that it has been ignored in the SDE algorithm; and its value is assumed as 1/4 of the total search space for each variable. As an example, if a precision of two digits is considered, then the value of this parameter will be equal to 2.

3.2 Steps of the SDE algorithm

Step 1. Creating [L]N×M matrix

This matrix is created by random number in the first loop, so that all the numbers are in the permissible range; and in the other loops, the L matrix is formed considering pervious loop as explained in the following sections.

Step 2. Computing PL1

Either $PL1$ can be set to 0.1 (10%), or it can be calculated using Eq. (11) after forming the matrix L for the first loop and generating Ci and $Modei$ for each variable.

Step 3. Calculating [fitness]N×2

In order to calculate the fitness, first the L matrix should be ordered, then fitness matrix that has two columns should be calculated so that the first column is the number of rows of ordered L matrix, and the second column is the fitness relative to every corresponding rows of L matrix. Ordering L matrix before calculating fitness is a good idea in this algorithm, because this process has a high impact on decreasing the number of loops and increasing the accuracy of the final results. This important change applied to the SDE algorithm and further explanations will be provided in the next steps. At the end of this step, all variables will be evaluated in few processes until the information is obtained for the subsequent loop.

Step 4. Calculating accumulative fitness, AF matrix

Each entry in $[L]_{N\times M}$ corresponds to one point in the coordinate system against their own fitness. The algorithm assumes a triangle distribution on the left and right part of each variable in each location. The length of the distribution is 2×R, where R is the effective radius as shown schematically in Fig. 1.

Overlapping usually happens in some parts of distribution. The more sub-curved area in a part, the probability of choosing that part as answer will be more than other parts. Thus accumulative fitness being shown by AF is calculated. AF takes overlaps and adds them for every alternative.

Alternatives matrix, which is one of the parameters in this algorithm, is omitted. This matrix was applied in the DE algorithm because the entire search space should be numbered. In this matrix the entire search space is ordered upwardly and the DE algorithm evaluates L matrix numbers by exploring in the Alternatives matrix. Thus numbering will begins from 1 to $(b\text{-}a)/AC$+1 for each variable, where a and b are the starting and ending points of the interval search space, respectively. Therefore, in the SDE algorithm instead of making the Alternative matrix, the following equation is utilized

$$A_{i,j} = \frac{L_{i,j} - a}{AC} + 1 \tag{13}$$

This change makes the burden of calculations to be reduced considerably, because generating alternative matrix and finding the elements of L matrix takes a lot of time. Thus, the Alternative matrix is omitted from the SDE algorithm. The SDE algorithm performs this process for each variable independently. As mentioned, the distributions are triangular. On occasion, some of these distributions were out of the range. To solve this problem, the borders were assumed to act as mirrors and reflect those parts into the borders. Fig. 2 shows (1) and (2) distributions reflected inward as (1)' and (2)'. Therefore, the algorithm adds these values to primary values of the AF. Thus, the SDE algorithm finds the row of L matrix which has the most fitness. In fact, this answer is the best one among the search space on that loop. AF is equaled to PLi for this location and the probability of $1-PLi$ is divided between the parts of the search space. This step is a repetitive process for the all variables. Sometimes, there is more than one maximum, including in the last loops. The SDE algorithm has predicted this issue and dispenses the PLi among them in proportion to their repetitions.

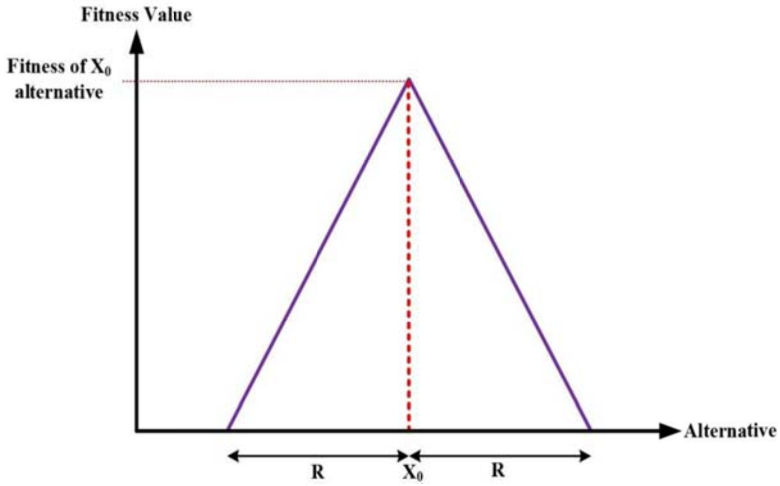

Fig. 1 A triangle distribution for the fitness function (Kaveh and Hosseini 2014)

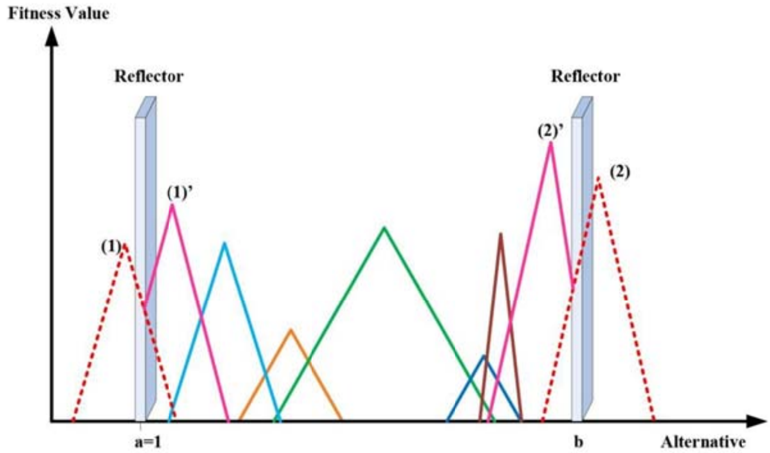

Fig. 2 The triangle distribution and their overlaps and reflections (Kaveh and Hosseini 2014)

Step 5. Generating $AFArea_{ij}$

The *AF* sub-curved area should be one or 100%, which is evaluated by the following equation

$$AFArea_{ij} = \frac{Area_{ij}}{\sum_{i=1}^{N} Area_{ij}} \tag{14}$$

where $AFArea_{ij}$ and $Area_{ij}$ are the modified AF sub-curved area and the AF sub-curved area, respectively. Now, $[AFArea]_{N\times 2}$ is generated in which sub-curved area is equal to one (100%). The DE algorithm uses Eq. (14) for getting at this aim (Kaveh and Farhoudi 2013).

$$P_{ij} = \frac{AF_{ij}}{\sum_{i=1}^{LA} AF_{ij}} \tag{15}$$

However, there is no guarantee for sub-curved area to be one with equation.

Step 6. Creating [L]N×M matrix for the next loop

In this section, the algorithm should generate L matrix for the next loop as an answer for the existing loop. There are different ways to perform this and the SDE algorithm has chosen a simple way for it. As the sub-curved area is one, the algorithm generates new locations by increasing the sub-curved area for a given percent value. This given value is *100% / M*. As this value increases, the SDE algorithm chooses a new member from *AFAreaij*. Performing this process, all the locations will be chosen. As mentioned in step 3, after performing all the steps independently for each variable, the algorithm should start some process for integrating variables. In fact the integrating action is for putting the best answers together that are selected from *AFAreaij* for every variable. It is not expected the appropriate answers to lie beside each other in one location; this can be modeled in dolphin's brain. Receiving the information, the dolphin thinks about them and then decides on next movement. Therefore, the *L* matrix should be ordered, and this ordering is the most important change on DE algorithm. Before calculating the fitness, the SDE algorithm uses some simple ways for ordering the *L* matrix as follows:

For the *j*th variable, constant values are considered for other variables and then fitness is calculated. Thus the *j*th variable is obtained according to the fitnesses ordered upwardly. This process is performed for all the variables. Now, the last location is the appropriate answer in the search space. Besides, the algorithm uses another way alongside with this one in which it combines different locations together randomly and makes some new combinations. Also, the algorithm is given a mutation ability, which means that the L matrix is allocated an allowed random number in 30% of time. It is preferred the ordering operation to be done in the first loop of L matrix too to increase the rate of convergence. These steps are repeated from step 1 to step 6 according to the loops numbers. For further clarity the flow chart of the algorithm is provided in Fig. 3.

4. Numerical examples

In this section four numerical examples and their results are provided. To demonstrate the ability of the algorithm four planar and spatial trusses with some single and multiple scenarios are studied. The scenarios are selected by increasing the number of free noise damaged elements and changing severity of the damages in the scenarios until the exact answer is not reached within the number of loops considered for one scenario. Then the scenarios obtained are tested by applying two different noises. For the first noise, *Noisef* and *Noiseφ* are 1% and 3%, respectively, and for the second noise these are 2% and 5%, respectively. In all of the tables and figures in this paper, * and † denote the first and second noise, respectively. In addition, all the scenarios of examples are run only few times. The way for obtaining the exact answer for each scenario is illustrated in each example. If the exact answer is found in scenarios within few runs, then the loop number in which the exact answer is achieved, is presented; otherwise, the result found in the last loop within few runs is provided.

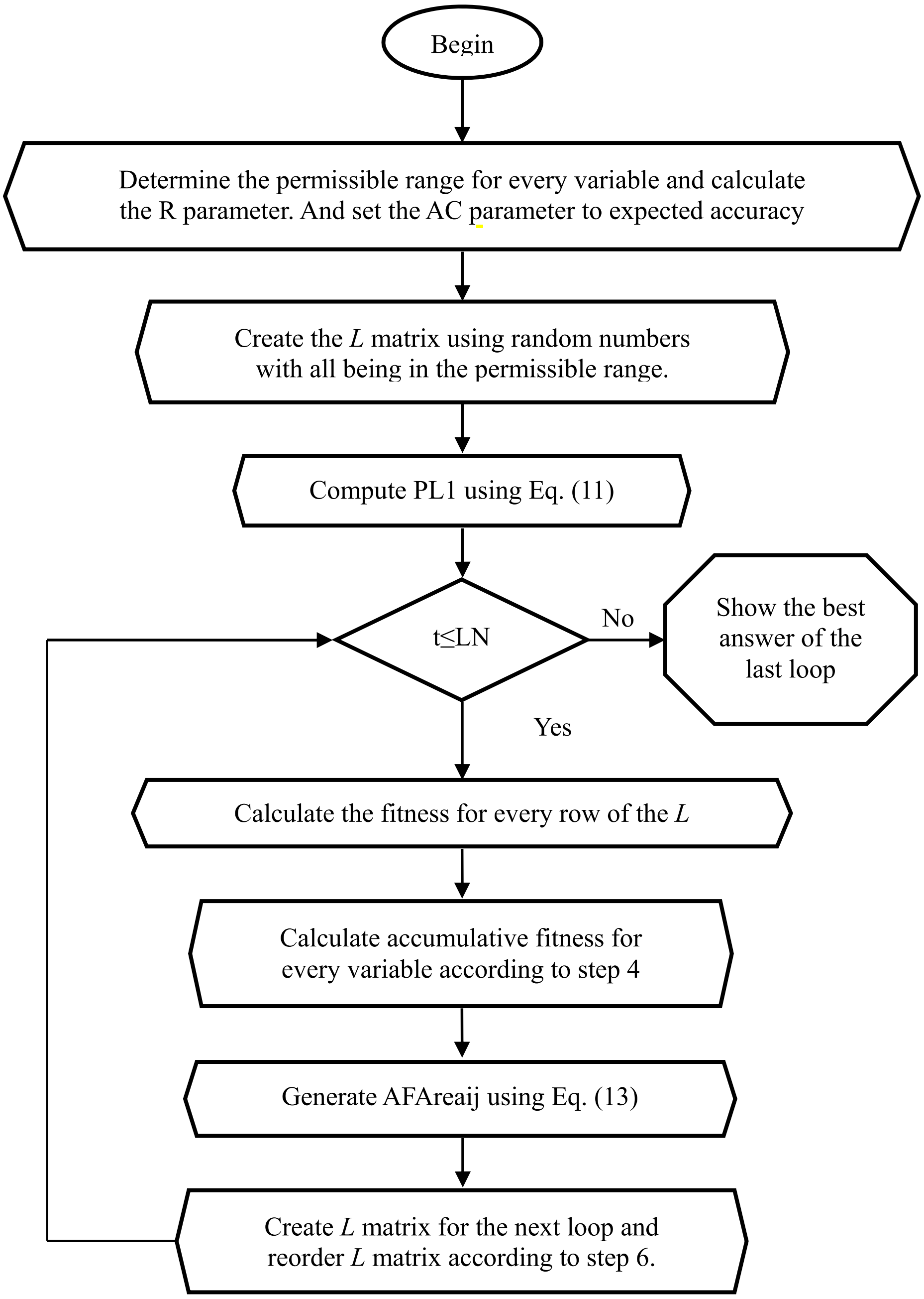

Fig. 3 Flowchart of the SDE

For a better understanding of the results, the misidentified elements found are given in all the tables. The number of the modes and the degrees of freedom used as input data in each structure are very important and should carefully be chosen. The more number of input data, the getting correct answer becomes easier by the algorithm. But in that case the rate of program running decreases and for obtaining the answer, the algorithm needs more time. It is advisable to generate ideal balance for algorithm by selecting an appropriate number of input data. The number of modes considered for the structures will be mentioned in each example, meanwhile, the number of degrees of freedom in each problem is the total number of degrees of freedom involved in each mode. Also, the value of AC algorithm parameter is assumed to be 0.1 (10%). For first, second, third and fourth examples, the number of loops are considered as 50, 50, 50 and 130; and the population sizes are assumed as 50, 50, 70 and 350, respectively.

4.1 A 10-bar planar truss

The first example is a 10-bar planar truss as shown in Fig. 4. Many researchers (Kaveh and Zolghadr (2011, 2012), Pholdee and Bureerat (2014), Lingyun *et al.* (2005) among others) have considered this truss as a benchmark in the field of optimization. This truss has 8 degrees of freedom and a non-structural mass of 454.0 kg is added to the free nodes. All of the elements in the structure have a modulus of elasticity E = 6.98×1010 N/m^2, density ρ = 2770 kg/m^3, and cross- sectional area A = 0.0025 m^2. All of the modes are considered (8 modes). Thirteen free noise damage scenarios are obtained for this example as shown in the Table 1.

All of the scenarios are run only few times and the loop numbers in the table imply the exact answer being achieved. Table 2 shows the results of the scenarios that are not reached the exact answer for this example.

Table 1 Damage scenarios of the 10-bar planar truss

Scenario	Damaged element(s)	Damage severity (β)	Loop number		
			Noise free	Noise*	Noise†
1	5	0.15	16	-	-
2	1	0.05	23	-	-
3	1,10	0.05,0.10	40	-	-
4	2,4	0.10,0.05	23	45	27
5	1,6,10	0.05,0.15,0.10	24	-	-
6	2,4,5	0.10,0.05, 0.15	36	21	-
7	2,3,4,5	0.10,0.40,0.05,0.15	21	25	-
8	2,3,4,5	0.10,0.20,0.05,0.15	24	25	21
9	2,3,4,5,10	0.10,0.20,0.05,0.15,0.10	23	32	-
10	3,5,8,9,10	0.30,0.12,0.22,0.05,0.20	29	34	-
11	2,3,4,5,7,9	0.10,0.40,0.05,0.15,0.10,0.15	21	-	-
12	1,3,4,5,7,9	0.10,0.40,0.15,0.15,0.21,0.15	24	-	-
13	2,4,5,6,7,9,10	0.10,0.25,0.15,0.05,0.10,0.15,0.15	-		

Table 2 Results of the 10-bar planar truss

Scenario	Damaged element(s)	Damage severity (β)
1*	1,3,5	0.01,0.01,0.15
1†	2,4,5,6,7,8,9,10	0.02,0.02,0.16,0.01,0.02,0.02,0.03,0.01
2*	1,9,10	0.05,0.01,0.01
2†	1,9,10	0.05,0.01,0.01
3*	1,7,8,10	0.05,0.01,0.01,0.11
3†	1,3,6,7,8,9,10	0.06,0.01,0.01,0.01,0.01,0.02,0.11
5*	1,6,7,8,9,10	0.05,0.15,0.01,0.01,0.01,0.10
5†	1,3,6,9,10	0.06,0.01,0.16,0.01,0.10
6†	1,2,3,4,5,6,7,8,10	0.03,0.10,0.02,0.05,0.16,0.01,0.01,0.01,0.01
7†	1,2,3,4,5,6,7,8,9	0.02,0.10,0.41,0.06,0.16,0.01,0.01,0.01,0.01
9†	2,3,4,5,8,9,10	0.08,0.19,0.05,0.15,0.02,0.01,0.09
10†	1,2,3,4,5,6,8,9,10	0.01,0.01,0.31,0.01,0.14,0.02,0.21,0.07,0.22
11*	2,3,4,5,7,9,10	0.10,0.40,0.05,0.15,0.10,0.16,0.01
11†	1,2,3,4,5,7,8,9,10	0.02,0.09,0.41,0.05,0.15,0.11,0.01,0.17,0.02
12*	1,3,4,5,7,9	0.10,0.40,0.15,0.15,0.21,0.16
12†	1,3,4,5,7,9	0.10,0.40,0.15,0.15,0.21,0.14
13	2,4,5,6,7,9,10	0.07,0.22,0.13,0.03,0.10,0.13,0.12

Adding noise in a problem causes errors in the experimental modal data. Thus there is no real scenario corresponding to these data. Therefore, the algorithm finds the scenario with its modal data having the least difference with experimental modal data. This scenario is or is not the exact answer, depends on the amount of noise, structure characteristics and assumed scenario.

Since there are rather many scenarios in the examples, one of the scenarios for which the exact answer is reached in all its conditions, is selected for representing variation of the objective function in each example.

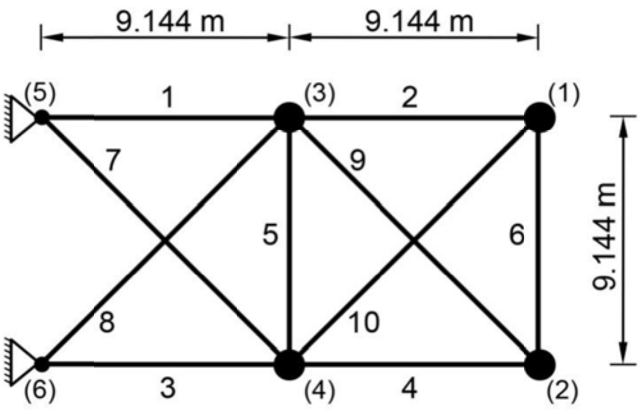

Fig. 4 A 10-bar planar truss

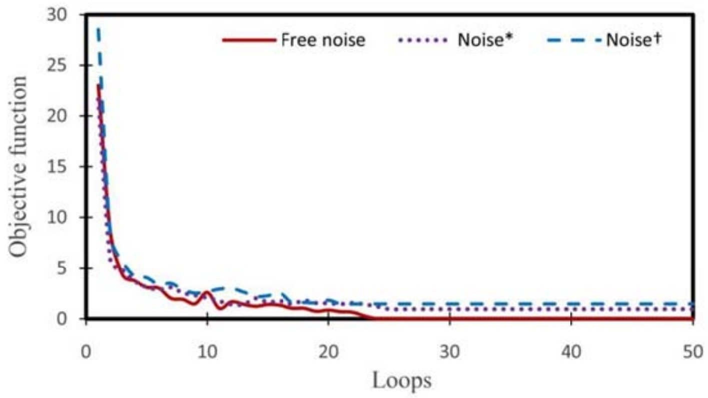

Fig. 5 Variation of the objective function with the number of loops for the 10-bar truss (Scenario 8)

Fig. 5 shows the variation of the objective function with the number of loops for scenario 8.

Despite finding the exact answer in both conditions, with and without noise in the scenario 8, considering Fig. 5, the value of objective function is not zero. In the case with noise condition this error in objective function is due to the noise.

It should be noted that first and last scenarios were computed in 10 independent iterations. The first scenario reaches to the exact answer in all 10 iterations for all cases, but the last scenario in noise free case reaches the exact answer in 2 iterations and other results have acceptable values. Mean values for every case are 1.03, 1.75 and 2.17 for noise free, Noise* and Noise†, respectively.

4.2 A 15-bar planar truss

The second example is a 15-bar planar truss as shown in Fig. 6. Villalba and Laier (2012, 2014) have used this truss and assessed different damage scenarios. This truss has 13 degrees of freedom. All of the elements in the structure have a modulus of elasticity E = 200×109 N/m^2, density ρ = 7800 kg/m, and cross-sectional area A = 0.001 m^2; and vertical and horizontal elements have a length equal to 1.0m. All of the modes are considered (13 modes). Fifteen noise free damage scenarios are obtained for this example as indicated in the Table 3.

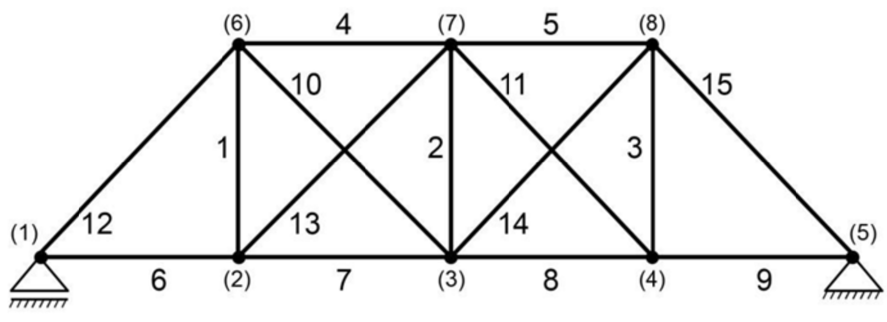

Fig. 6 A 15-bar planar truss.

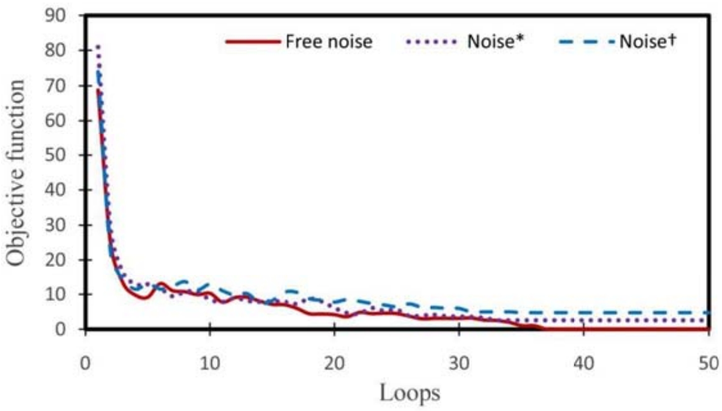

Fig. 7 Variation of the objective function with number of loops for the 15-bar truss (Scenario 13)

Table 3 Damage scenarios of the 15-bar planar truss

Scenario	Damaged element(s)	Damage severity (β)	Loop number		
			Noise free	Noise*	Noise†
1	7	0.18	25	25	26
2	13	0.33	24	23	32
3	6,11	0.20,0.15	35	30	31
4	2,7	0.20,0.10	36	26	-
5	1,7,13	0.47,0.25,0.30	27	24	-
6	2,6,11	0.16,0.20,0.20	30	46	29
7	1,6,7,11	0.47,0.20,0.25,0.20	50	31	27
8	2,6,7,13	0.16,0.20,0.25,0.30	45	27	31
9	2,6,7,10,13	0.16,0.20,0.25,0.17,0.30	39	41	28
10	3,4,8,11,14	0.16,0.23,0.10,0.30,0.18	39	39	-
11	2,5,9,12,13,14	0.15,0.12,0.10,0.05,0.10,0.24	44	35	-
12	3,6,8,10,11,15	0.13,0.12,0.14,0.20,0.10,0.24	36	38	35
13	3,6,7,9,12,13,15	0.20,0.16,0.17,0.13,0.14,0.06,0.08	37	33	35
14	3,6,8,10,11,13,15	0.17,0.14,0.14,0.23,0.14,0.16,0.11	39	37	-
15	3,6,8,10,11,12,13,15	0.17,0.14,0.14,0.23,0.14,0.21,0.16,0.	-		

Table 4 Results for the 15-bar planar truss

Scenario	Damaged element(s)	Damage severity (β)
4†	2,4,7,10,11	0.21,0.01,0.10,0.01,0.01
5†	1,7,9,13	0.47,0.25,0.01,0.30
10†	3,4,8,11,14	0.15,0.23,0.09,0.30,0.18
11†	2,5,9,10,12,13,14	0.15,0.12,0.08,0.01,0.05,0.09,0.24
14†	3,6,8,10,11,13,15	0.17,0.14,0.14,0.23,0.13,0.16,0.11
15	3,6,8,10,11,12,13,15	0.17,0.14,0.14,0.24,0.14,0.21,0.15,0.11

Table 4 shows the results of the scenarios that are not reached the exact answer for this example. Fig. 7 shows the variation of objective function with the number of loops for scenario 13.

4.3 A 25-bar spatial truss

The third example is a 25-bar spatial truss as shown in Fig. 8. Kim *et al*. (2014) have used this truss and detected damage using a two-stage optimization. This truss has 10 nodes and 18 degrees of freedom. All of the elements in the structure have a modulus of elasticity $E = 10$ GPa, density $\rho = 0.1$ kg/m^3, and cross-sectional area of $A = 0.01$ m^2. All of the modes are considered (18 modes).

Eleven free noise damage scenarios are obtained for this example are shown in the Table 5. In the scenario 11, damage severity for elements 2, 7, 10, 13, 17 and 21 are computed equal to 0.19, 0.13, 0.22, 0.13, 0.15 and 0.17, respectively.

Fig. 9 shows the variation of objective function with the number of loops for scenario 10.

Table 6 shows the results of the scenarios of the 25-bar spatial truss by considering the seven first modes. Only one scenario reaches the exact answer in Noise† scenarios of this case. This is due to the effect of the larger value of Noise† and considering only the seven first modes. In the case of Noise†, since the value of the objective function for the exact magnitude of damage is more than the value found by the SDE method, and the algorithm tries to find a value less than the objective function, therefore finding an exact answer become impossible.

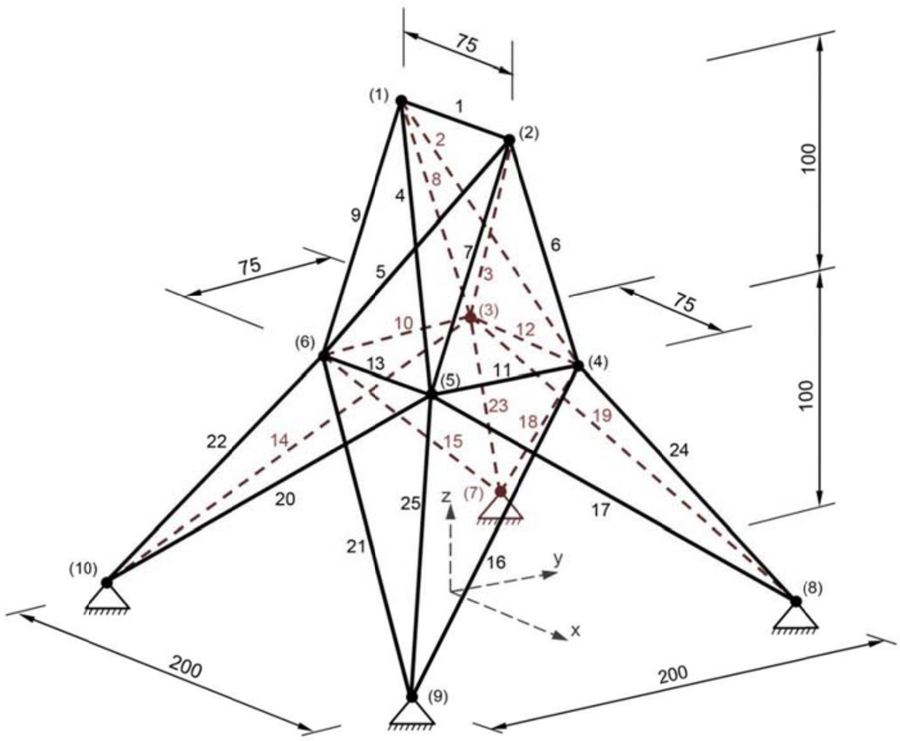

Fig. 8 A 25-bar spatial truss (dimensions are in mm)

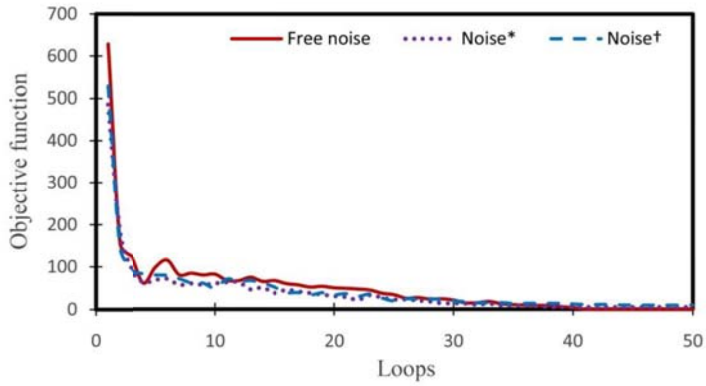

Fig. 9 Variation of the objective function with number of loops for the 25-bar truss (Scenario 10)

Table 5 Damage scenarios of the 25-bar spatial truss

Scenario	Damaged element(s)	Damage severity (β)	Loop number		
			Noise free	Noise*	Noise†
1	19	0.18	36	35	38
2	9	0.16	34	35	37
3	7,23	0.15,0.20	39	35	37
4	4,11	0.20,0.10	43	36	50
5	2,10,18	0.20,0.25,0.15	41	35	36
6	7,17,23	0.15,0.15,0.20	41	41	37
7	2,7,13,22	0.20,0.14,0.13,0.17	38	48	48
8	1,3,8,20	0.15,0.20,0.15,0.20	45	48	45
9	2,5,10,18,24	0.15,0.20,0.20,0.15,0.20	50	50	47
10	2,6,12,17,21	0.20,0.14,0.13,0.15,0.17	41	38	45
11	2,7,10,13,17,21	0.20,0.14,0.22,0.13,0.15,0.17	-		

Table 6 Damage scenarios of the 25-bar spatial truss (by considering the seven first modes)

Scenario	Damaged element(s)	Damage severity (β)	Loop number		
			Noise free	Noise*	Noise†
1	19	0.18	32	43	-
2	9	0.16	31	41	36
3	7,23	0.15,0.20	46	40	-
4	4,11	0.20,0.10	37	49	-
5	2,10,18	0.20,0.25,0.15	47	36	-
6	7,17,23	0.15,0.15,0.20	45	40	-
7	2,7,13,22	0.20,0.14,0.13,0.17	50	47	-
8	1,3,8,20	0.15,0.20,0.15,0.20	44	42	-
9	2,5,10,18,24	0.15,0.20,0.20,0.15,0.20	40	45	-
10	2,6,12,17,21	0.20,0.14,0.13,0.15,0.17	47	41	-
11	2,7,10,13,17,21	0.20,0.14,0.22,0.13,0.15,0.1	-		

4.4 A 72-bar spatial truss

The fourth example is a 72-bar spatial truss that is shown in Fig. 10. Many researchers (Kaveh and Zolghadr (2015, 2011, 2012), Sedaghati (2005), Konzelman (1986) and Gandomi (2014) among others) have considered this truss as benchmark in the field of optimization. This truss has 20 nodes and 48 degrees of freedom, and four non-structural masses of 2270.0 kg are added to the nodes 1-4. All of the elements in the structure have a modulus of elasticity E = 6.98×1010 N/m^2, density ρ = 2770 kg/m^3, and cross-sectional area A = 0.0025 m^2. The sixteen first modes are considered. Five noise free damage scenarios are obtained for this example as indicated in Table 7.

Table 8 shows the results of the scenarios which have not reached the exact answer for this example.

Fig. 11 shows the variation of the objective function with the number of loops for scenario 3.

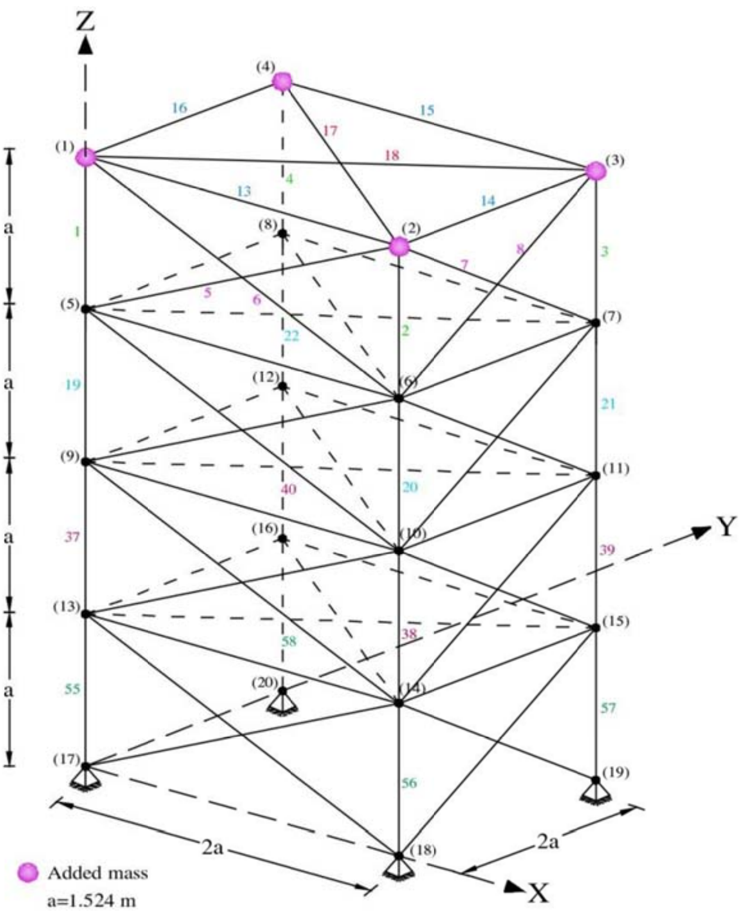

Fig. 10 A 72-bar spatial truss

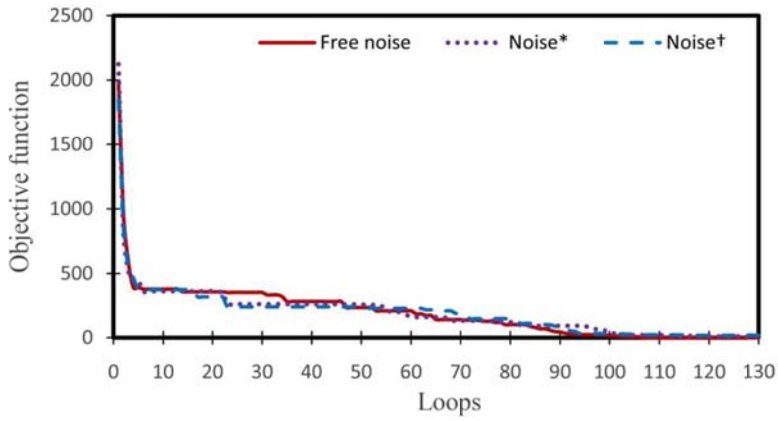

Fig. 11 Variation of the objective function with number of loops for the 72-bar truss (Scenario 3)

Table 7 Damage scenarios of the 72-bar spatial truss

Scenario	Damaged element(s)	Damage severity (β)	Loop number		
			Noise free	Noise*	Noise†
1	10	0.20	103	-	-
2	55	0.15	100	106	-
3	4,58	0.10,0.15	108	120	116
4	14,51	0.13,0.10	108	121	-
5	4,14,58	0.10,0.13,0.15	-		

Table 8 Results for the 72-bar spatial truss

Scenario	Damaged element(s)	Damage severity (β)
1*	10,69	0.20,0.01
1†	10,69,70	0.20,0.01,0.01
2†	55,68	0.15,0.01
4†	14,51	0.13,0.11
5	4,14,22,27,35,40,45,53,56,58,68,70	0.09,0.13,0.01,0.01,0.01,0.04,0.01,0.01, 0.01,0.11,0.01,0.02

4.4.1 Assigning a significant mutation for damage detection

In studying damage detection problems, it is clear that most of the members are undamaged and the assigned numbers are equal to zero and few of them are damaged with non-zero numbers. Considering this issue, it is possible to give an opportunity to the algorithm to obtain the exact answer in a much shorter time than mode. Thus, it is possible to generate a significant mutation in the algorithm, so that it induces zero to answers emphatically. To do this, the following formula is used

```
for i=1:N
for j=1:M
if rand<0.3
L(i,j)=0
End
End
End
```
(16)

where *rand* is a random number generated by MATLAB. This formula includes constant value equal to zero to 30% of the answers. Then, the scenarios relevant to this example are solved by this mutation with the results as indicated in Table 9.

It should be mentioned that the number of loops and population are set to 50.

As the results show, using this mutation with fewer loops and smaller population size provides better results.

Table 10 shows the results of the scenarios that are not reached the exact answer for this example with mutation.

Fig. 12 shows the variation of the objective function with the number of loops for scenario 3.

Table 9 Damage scenarios of the 72-bar spatial truss with mutation

Scenario	Damaged element(s)	Damage severity (β)	Loop number		
			Noise free	Noise*	Noise†
1	10	0.20	50	-	50
2	55	0.15	45	46	46
3	4,58	0.10,0.15	49	50	49
4	14,51	0.13,0.10	50	-	-
5	4,14,58	0.10,0.13,0.15	48	-	-
6	3,16,56	0.12,0.15,0.13	50	-	-
7	4,14,45,58	0.10,0.13,0.13,0.13	-		

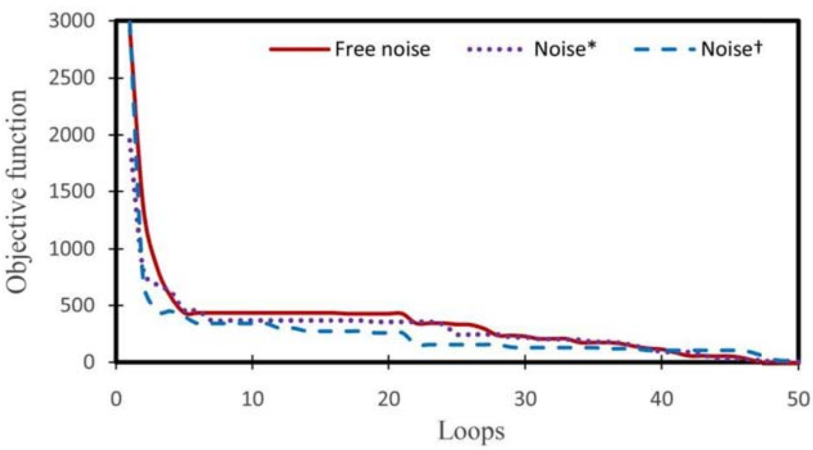

Fig. 12 Variation of the objective function with number of loops for the 72-bar truss with mutation (Scenario 3)

Table 10 Results for the 72-bar spatial truss with mutation

Scenario	Damaged element(s)	Damage severity (β)
1*	10,69	0.20, 0.01
4*	14,50,51	0.13,0.01,0.07
4†	14,51,68	0.13,0.02,0.01
5*	4,14,36,58	0.10,0.13, 0.01,0.14
5†	4,14,40,58,67	0.08,0.12,0.01,0.13,0.01
6*	3,16,23,41,56	0.12,0.08,0.01,0.01,0.11
6†	3,16,33,56	0.12,0.15,0.01,0.14
7	4,14,45,51,58	0.10,0.13,0.13,0.01,0.13

Table 11 Results of the 10-bar planar truss corresponding SDE and Kaveh and Mahdavi (2016)

Scenario		Damaged element (s)	Damage severity (β)
1	Exact	1	0.05
	SDE (best and mean)	1	0.05
	CBO (best)	1	0.05
	ECBO (best)	1	0.05
	CBO (mean)	1	0.04916
	ECBO (mean)	1	0.04977
2	Exact	2,4	0.10,0.05
	SDE (best and mean)	2,4	0.10,0.05
	CBO (best)	2,4	0.10,0.051
	ECBO (best)	2,4	0.10,0.05
	CBO (mean)	2,4	0.0811,0.0598
	ECBO (mean)	2,4	0.10,0.0499

Table 12 Results of the 15-bar planar truss corresponding SDE and Villalba and Laier (2012)

Scenario		Damaged element (s)	Damage severity (β)
1	Exact	7	0.18
	SDE	7	0.18
	multi-chromosome GA	7,8	0.174,0.039
2	Exact	13	0.33
	SDE	13	0.33
	multi-chromosome GA	13	0.333
3	Exact	1,7,13	0.47,0.25,0.30
	SDE	1,7,13	0.47,0.25,0.30
	multi-chromosome GA	1,7,13	0.472,0.253,.309
4	Exact	2,6,11	0.16,0.20,0.20
	SDE	2,6,11	0.16,0.20,0.20
	multi-chromosome GA	2,6,11	0.168,0.20,0.203

Table 13 Results of the 72-bar spatial truss corresponding SDE and Kaveh and Mahdavi (2016)

Scenario		Damaged element(s)	Damage severity (β)
1	Exact	55	0.15
	SDE (best and mean)	55	0.15
	CBO (best)	55	0.15
	ECBO (best)	55	0.15
	CBO (mean)	55	0.1348
	ECBO (mean)	55	0.1457
2	Exact	4,58	0.10,0.15
	SDE (best and mean)	4,58	0.10,0.15
	CBO (best)	4,58	0.0989,0.15
	ECBO (best)	4,58	0.10,0.15
	CBO (mean)	4,58	0.0916,0.1515
	ECBO (mean)	4,58	0.1017,0.1515

5. Conclusions

5.1 Comparison of the algorithm efficiency with other works

The scenarios shown in Tables 11 to 13 are selected in order to compare the efficiency of the present algorithms with the two studies mentioned for three examples including the 10-bar planar truss, the 15-bar planar truss and the 72-bar spatial truss. Thus the scenarios mentioned are simulated in accordance with the details of the studies and the results obtained by the SDE are exact for all of the scenarios.

5.2 Discussion on noise free condition

The scenarios are selected by increasing the number of damaged elements and changing the severity of damaged elements in the scenarios until finding the exact answer within the assumed number of loops and a few runs becomes impossible. In other words, the number of loops and running only a few times are two important conditions here. This means the answer presented for the last scenario in each example is not the real answer, because considering these conditions the algorithm has been unable to get the exact answer. Therefore it is impossible to claim that the algorithm has trapped in a local minimum. On the other hand, given the results of these scenarios, the locations of damaged elements are detected correctly and only few misidentified elements are found which in fact are not real, because if the algorithm was given enough chance or the number of runs or assumed population were higher, it would have been possible to achieve the exact detection.

5.3 Discussion on added noise condition

The errors in the computation of damage severity for the real damaged elements were 0.02, 0.02 and 0.01 in scenarios of first, second and last trusses, respectively; and for the last truss with mutation, this error was 0.08 in scenario 4† that is rather high, but this value was reduced to zero after three other runs. There were maximums 7, 3 and 2 misidentified elements with damage severity less than or equal to 0.03 among scenarios of the first, second and fourth examples, respectively. For the last example with mutation, there were maximum 2 misidentified elements with damages equal to 0.01 in scenarios 5† and 6*. Generally, simulating identical damage scenarios with two different noise levels showed that the errors in the damage severity of real damage elements and the number and extent damage of misidentified elements were increased as the noise increased.

5.4 Conclusion

In this paper the application of the Simplified Dolphin Echolocation to the damage detection problem is investigated. The objective function is formulated based on natural frequencies and mode shapes. Therefore, detection of damage in truss structures was addressed by solving inverse optimization problem. By examining different truss structures, it is found that the algorithm is capable of detecting damage of different multiple scenarios with and without noise in a few runs. Also, a new mutation is proposed for damage detection problem in the last example which uses this mutation with smaller size population and a much shorter computational time leading to better result in a few runs. Incorporating this mutation with other algorithms like PSO, ACO, ICA, CSS, CBO etc. for damage detection in different truss structures may also lead to desirable results with a smaller population size and less number of loops with a shorter computational time.

Acknowledgments

The first author is grateful to the Iran National Science Foundation for the support.

References

Carden, E.P. and Fanning, P. (2001), "Vibration based condition monitoring: a review", *Struct. Health Monit.*, **3**(4), 355-377.

Chen, B. and Nagarajaiah, S. (2013), "Flexibility-based structural damage identification using Gauss–Newton method", *Sadhana*, **38**(4), 557-569.

Doebling, S.W., Farrar, C., Prime, M.B. and Shevitz, D.W. (1996), *Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics a literature review*, Los Alamos National Lab, NM, United States.

Doebling, S.W., Farrar, C.R. and Prime, M.B. (1998), "A summary review of vibration-based damage identification methods", *Shock Vib. Digest*, **30**(2), 91-105.

Fan, W. and Qiao, P. (2011), "Vibration-based damage identification methods: a review and comparative study", *Struct. Health Monit.,* **10**(1), 83-111.

Gandomi, A.H. (2014), "Interior search algorithm (ISA): a novel approach for global optimization", *ISA Trans.*, **53**(4), 1168-1183.

Hakim, S.J.S. and Abdul Razak, H. (2013), "Structural damage detection of steel bridge girder using artificial neural networks and finite element models". *Steel Compos. Struct.*, **14**(4) , 367-377.

hybridized CSS-BBBC algorithm with trap recognition capability", *Comput. Struct*., **102**, 14-27.

Kaveh, A. (2014), "*Advances in metaheuristic algorithms for optimal design of structures*", Springer International Publishing, Switzerland.

Kaveh, A. and Farhoudi, N. (2013), "A new optimization method: Dolphin echolocation", *Adv. Eng. Softw*., **59**, 53-70.

Kaveh, A. and Farhoudi, N. (2016), "Dolphin echolocation optimization: continuous search space", *Adv. Comput. Des*., **1**(2), 175-194.

Kaveh, A. and Hosseini, P. (2014), "A simplified dolphin echolocation optimization method for optimum design of trusses", *Int. J. Optim. Civil Eng*., **4**(3), 381-397.

Kaveh, A. and Mahdavi, V. (2016), "Damage Identification of truss structures using CBO and ECBO algorithms", *Asian J. Civil Eng. (BHRC)*, **17**(1), 75-89.

Kaveh, A. and Maniat, M. (2014), "Damage detection in skeletal structures based on charged system search optimization using incomplete modal data", *Int. J. Civil Eng*., *IUST*, **12**(2), 193-200.

Kaveh, A. and Maniat, M. (2015), "Damage detection based on MCSS and PSO using modal data", *Smart Struct. Syst*., **15**(5)**,** 1253-1270.

Kaveh, A. and Zolghadr, A. (2011), "Shape and size optimization of truss structures with frequency constraints using enhanced charged system search algorithm", *Asian J. Civil Eng.*, **12**(4), 487-509.

Kaveh, A. and Zolghadr, A. (2012), "Truss optimization with natural frequency constraints using a hybridized CSS-BBBC algorithm with trap recognition capability", *Comput. Struct*., **102**, 14-27.

Kaveh, A. and Zolghadr, A. (2015), "An improved CSS for damage detection of truss structures using changes in natural frequencies and mode shapes", *Adv. Eng. Softw*., **80**, 93-100.

Kim, N., Kim, H. and Lee, J. (2014), "Damage detection of truss structures using two-stage optimization based on micro genetic algorithm", *J. Mech. Sci. Technol*., **28**(9), 3687-3695.

Konzelman, C.J. (1986), Dual methods and approximation concepts for structural optimization.

Kwon, Y.W. and Bang, H. (2000), *The finite element method using MATLAB,* CRC Press.

Lingyun, W. *et al*. (2005), "Truss optimization on shape and sizing with frequency constraints based on genetic algorithm", *Comput. Mech*., **35**(5), 361-368.

Majumdar, A., Maiti, D.K. and Maity, D. (2012), "Damage assessment of truss structures from changes in natural frequencies using ant colony optimization", *Appl. Math. Comput*., **218**(19), 9759-9772.

Pan C.D., Yu L., Chen Z.P., Luo W.F. and Liu H.L. (2016). "A hybrid self-adaptive Firefly-Nelder-Mead algorithm for structural damage detection". *Smart Struct. Syst., An Int'l Journal*, 17(6), 957-980.

Pholdee, N. and Bureerat, S. (2014), "Comparative performance of meta-heuristic algorithms for mass minimisation of trusses with dynamic constraints", *Adv. Eng. Softw*., **75**, 1-13.

Sedaghati, R. (2005), "Benchmark case studies in structural design optimization using the force method", *Int. J. Solids Struct.*, **42**(21), 5848-5871.

Sohn, H. *et al*. (2004), *A review of structural health monitoring literature: 1996-2001*, Los Alamos National Lab, NM, United States.

Villalba-Morales, J.D. and Laier, D.K. (2014), "Assessing the performance of a differential evolution algorithm in structural damage detection by varying the objective function", *Dyna*, **81**(188), 106-115.

Villalba, J. and Laier, J.E. (2012), "Localising and quantifying damage by means of a multi-chromosome genetic algorithm", *Adv. Eng. Softw.*, **50**, 150-157.

CC