Arab J Sci Eng (2017) 42:1251–1263
DOI 10.1007/s13369-016-2398-6

RESEARCH ARTICLE - CIVIL ENGINEERING

Damage Detection of Thin Plates Using GA-PSO Algorithm Based on Modal Data

Seyed Rohollah Hoseini Vaez[1] · **Narges Fallah**[1]

Received: 1 July 2016 / Accepted: 14 December 2016 / Published online: 30 December 2016

Abstract In this study, a new approach for detecting damage, its location, and its severity in plate structures using a genetic–particle swarm optimization, which is a hybrid algorithm, is presented. To evaluate the proposed approach, three numerical examples have been simulated; the examples consist of three different plates including an L-shaped two-clamped supported plate, one quarter of a plate with a hole, and a rectangular two-clamped plate. These plate structures have been modeled using thin plate theory, so they are called thin plate. Additionally, dynamic method based on modal data such as natural frequencies and mode shapes is used to formulate objective function. In order to demonstrate the effectiveness of the new proposed approach and the hybrid algorithm, several structures are tested by several different scenarios with and without noise. Then, the scenarios are simulated with genetic and particle swarm optimization algorithms separately. Finally, the obtained results are compared using two sum error indexes which reveal that the results of the hybrid algorithm have less error.

Keywords Damage detection · Hybrid algorithm · GA-PSO · Thin plate · Modal data · Inverse problem

1 Introduction

Changes which take place within utilizing the structure are called damage. Damage detection is attributed to all methods and procedures which survey the damage, its location, and its quantity. Hitherto, several methods for detecting damage in structures have been proposed by researchers. Optimization algorithms and solving inverse optimization problem based on dynamic modal features are among common approaches. It is clear that damage brings about changes in structural properties including mass and stiffness or energy dissipation in the structure [1,2]. Thereupon, considering that a structure's modal parameters, such as natural frequencies and mode shapes, are subjected to these properties, they can be suitable criteria for formulating damage objective function. The purpose of formulating the objective function and solving optimization problem is to minimize the differences between dynamic parameters gained from experimental test and parameters gained from finite element model which represents the damaged structure. Through review of damage identification methods based on dynamic parameters is presented in [3–5].

Among the structures, plates are among the most important seismic-resistant components. Hence, evaluating their conditions is one of the most fundamental aspects of structural health monitoring.

Rytter [6] has defined four levels of damage detection as follows:

- Level 1: Determination of existence of damage in structure
- Level 2: Level 1 plus determination of the geometric location of the damage
- Level 3: Level 2 plus quantification of the damage severity
- Level 4: Level 3 plus prediction of the structure's remaining service life

Lately, many researchers have extensively begun to study damage detection in plates using techniques based on modal data. Most researches on detection of a plate's damage or

✉ Seyed Rohollah Hoseini Vaez
hoseinivaez@qom.ac.ir

[1] Department of Civil Engineering, Faculty of Engineering, University of Qom, Qom, Iran

crack can be categorized into two main methods. While the first methods detect the existence of damage and its location (level 2), the second methods not only detect the existence and location of damage, but estimate its severity as well (level 3). Thus, the latter is more powerful than the former.

Some of the studies applying the first method are as follows:

To assess damage location in plate structures, Bagheri et al. [7] have introduced a new method based on curvelet transform Curvelet transform is used because of its desirable performance in detecting line feature. Wang and Deng [8] have studied two-dimensional problems such as steel plate with a hole subjected to uniform tensile loading. This paper detects the location of damage by using Haar wavelet coefficient and applying static displacement as input of wavelet transform. Using wavelet analysis, a method for online detection of initial damage based on energy change of structural dynamic responses decomposed in composite structures has been proposed by Yan and Yam [9]. Rucka and Wilde [10] estimated the damage location in beam and plate structures using wavelet transform. They illustrated the location of damage with a peak in the spatial change of the transformed response. Kim et al. [11] have developed a reference-free impedance method for crack detection in a plate-like structure. In the study, the proposed technique uses a signal pair of PZTs arranged on the opposite surfaces of a structure to determine mode conversion produced with crack formation. For identifying the location and approximate shape or area of the damage in plate structures, Fan and Qiao [12] presented a new method based on two-dimensional continuous wavelet transform. A new damage indicator based on modal data for damage detection in plate-like structures is presented in [13]. It uses modal data such as mode shapes, its derivatives, and simulated numerical examples with and without noise to evaluate the exact localization of different damage cases. Xiang et al. [14] presented a new method based on operating deflection shape (ODS) to identify damage locations in plate-like structures.

Some of the studies which employ the second methods are as follows:

Song et al. [15] proposed a new approach which composes a parameter subset selection process with damage functions. The study evaluates the damage, its location, and its severity in complex structures, e.g., a 2D wall. Applying colonial competition algorithm, Nicknam and Hosseini [16] studied two kinds of different plate structures. The first one is a two-story shear wall, and the other one is a four-fixed supported plate. By this way, they detect the damage, its location, and its severity. Masoumi et al. [17] using imperialist competitive algorithm (ICA) have defined new scenarios with different levels of noise. Using finite element method, he models three structures including a clamped-free steel beam, a 2D-truss, and a plate-type structure, which evaluate their procedure in detecting damage. He also simulated the tested scenarios by ICA with binary genetic, and particle swarm optimization algorithms.

Genetic algorithm and particle swarm optimization are optimization techniques that can be applied to solve the structural damage identification problem. These techniques and some other optimization techniques have been applied for detecting damage in different structures by many researchers including particle swarm optimization [18,19], genetic algorithm [20], multi-objective optimization [21], cuckoo optimization [22].

The most studies by the second methods have been investigated by metaheuristics algorithms. The main advantage of the methods as acknowledged in the literature is its high efficiency for quantifying damage severity (level 3). Also, the most studied plates are modeled by thick (mindlin) theory.

This paper studies the damage detection in thin plate structures using a new hybrid algorithm namely genetic–particle swarm optimization (GA-PSO) based on variation in natural frequencies and mode shapes.

The sections of this study are as follows: In Sect. 2, damage detection approach is presented. The hybrid algorithm GA-PSO is presented in Sect. 3. Numerical examples and comparing the algorithms' performance are studied in Sect. 4. Finally, conclusions are given in Sect. 5.

2 Damage Detection Approach

The main task of the presented damage detection approach is solving the optimization problem with objective function and based on dynamic parameters of structure. The solution to the inverse optimization problem is performed by applying the stages shown in Fig. 1.

2.1 Stage 1: Determine the Finite Element Model for Intact Structure

To provide finite element model, stiffness and mass matrices for each element should be calculated initially. Then mass matrix, **M**, and stiffness matrix, **K** are calculated using Eqs. (1) and (2) as follows:

$$\mathbf{K} = \sum_{i=1}^{\mathrm{NE}} \mathbf{k}_i \tag{1}$$

$$\mathbf{M} = \sum_{i=1}^{\mathrm{NE}} \mathbf{m}_i \tag{2}$$

where $\mathbf{k}_i$ and $\mathbf{m}_i$ are the stiffness and mass matrices of ith element of the structure, respectively, and NE is the number of elements of the structure.

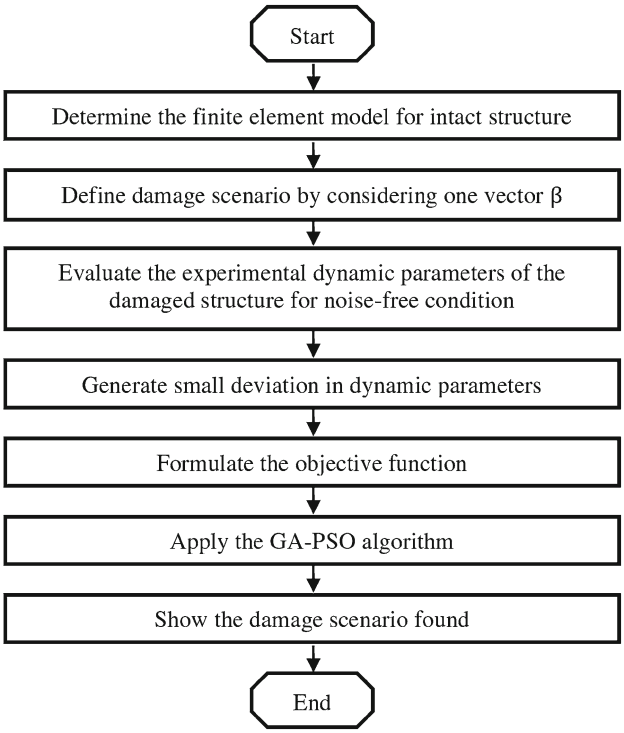

Fig. 1 Damage detection approach

In this study, the plate structures are modeled using thin plate theory, which is called thin plate. The plates are modeled with constant strain triangle (CST) elements and considering plane stress condition. Thus, stiffness and mass matrix for every typical element, shown in Fig. 2, is obtained by Eqs. (3) and (4) as follows:

$$\mathbf{k}^e = t_e A_e \mathbf{B}^T \mathbf{D} \mathbf{B} \tag{3}$$

$$\mathbf{m}^e = \frac{\rho t_e A_e}{12} \begin{bmatrix} 2 & 0 & 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 & 0 & 1 \\ 1 & 0 & 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 & 0 & 1 \\ 1 & 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 1 & 0 & 2 \end{bmatrix} \tag{4}$$

where t_e, ρ, and A_e are thickness, mass density, and area of the element, respectively. **B** and **D** are strain–displacement matrix of dimension 3×6 relating the three strains to the six nodal displacements and material matrix of dimension 3×3 for plan stress state, respectively, which can be stated as follows:

$$\mathbf{B} = \frac{1}{det\mathbf{J}} \begin{bmatrix} y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\ 0 & x_{32} & 0 & x_{13} & 0 & x_{21} \\ x_{32} & y_{23} & x_{13} & y_{31} & x_{21} & y_{12} \end{bmatrix} \tag{5}$$

$$\mathbf{D} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{6}$$

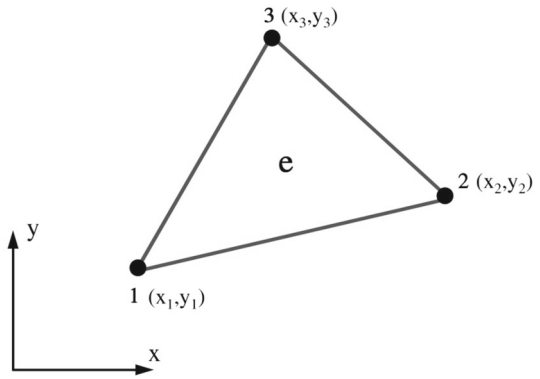

Fig. 2 Typical element

where x_{ij} and y_{ij} are $x_i - x_j$ and $y_i - y_j$, respectively. ν and E represent Poisson's ratio and elasticity modulus, respectively. **J** is Jacobian of the transformation matrix which can be expressed as follows:

$$\mathbf{J} = \begin{bmatrix} x_{13} & y_{13} \\ x_{23} & y_{23} \end{bmatrix} \tag{7}$$

2.2 Stage 2: Define Damage Scenario by Considering One Vector β

Typically, damage is modeled by reduction in structure's properties. In this study, however, the damage is identified using reduction in element's elasticity modulus. To model any damage in a structure, firstly a scenario should be defined. In fact, each considered scenario is a vector **β** which is a column vector of dimension $n \times 1$. Where n is the number of structure's elements. Therefore, numerical amount of ith array of this vector means the damage in ith element of structure. And the amount is between 0 and 1 for completely intact and damaged element, respectively. Thus, the relationship between the two conditions, damaged and intact, for ith element of the structure which is equal to:

$$E_{i\mathrm{d}} = (1 - \beta_{i\times 1}) \times E_{i\mathrm{h}} \tag{8}$$

where $E_{i\mathrm{d}}$ and $E_{i\mathrm{h}}$ are the elasticity moduli of the ith element for the damaged and intact conditions, respectively.

2.3 Stage 3: Evaluate the Experimental Dynamic Parameters of the Damaged Structure for Noise-Free Condition

Considering the supposed vector **β** in the previous stage, a reduction in the elasticity modulus induces a decrease in the stiffness of the structure based on:

$$\mathbf{K}_{\mathrm{d}} = \sum_{i=1}^{\mathrm{NE}} (1 - \beta_{i\times 1})\, \mathbf{k}_i \tag{9}$$

where $\mathbf{K}_{\mathrm{d}}$ is the stiffness matrix of the structure for damaged condition.

Finally, by adjusting the ith eigenvalue equation of damaged structure, dynamic parameters, considering the natural frequencies and mode shapes can be expressed as:

$$\left(\mathbf{K}_{\mathrm{d}} - \omega_{i\mathrm{d}}^2 \mathbf{M}\right) \boldsymbol{\varphi}_{i\mathrm{d}} = 0 \tag{10}$$

where $\omega_{i\mathrm{d}}$ and $\boldsymbol{\varphi}_{i\mathrm{d}}$ are the ith natural frequency and mode shape of the structure for damaged condition, respectively.

2.4 Stage 4: Generate Small Deviation in Dynamic Parameters

Avoiding the noise in real dynamic tests is impossible. Thereupon, by producing small deviation in the experimental dynamic parameters, this issue is measured as:

$$\omega_{\mathrm{noisy}} = \omega_{i\mathrm{d}} \times \left(1 + \alpha \times \mathrm{noise}_{\mathrm{freq}}\right) \tag{11}$$

$$\varphi_{ij\mathrm{noisy}} = \varphi_{i,j\mathrm{d}} \times (1 + \alpha \times \mathrm{noise}_{\mathrm{mode}}) \tag{12}$$

where noisy implies a noisy value. α is a uniformly distributed randomly number between -1 and +1. $\mathrm{noise}_{\mathrm{freq}}$ and $\mathrm{noise}_{\mathrm{mode}}$ denote deviations of the natural frequencies and mode shapes which are 1 and 3%, respectively [23,24].

2.5 Stage 5: Formulate the Objective Function

In this stage, using an objective function which formulated based on the natural frequencies and the mode shapes of the structure, the minimization of the optimization problem is solved. The objective function (F) used in this study is defined as:

$$F = \sqrt{\frac{1}{n}\left(\sum_{i=1}^{n}\left(\omega_i^{ex} - \omega_i^{gp}\right)^2 + \sum_{i=1}^{n}\sum_{j=1}^{nd}\left(\varphi_{ij}^{ex} - \varphi_{ij}^{gp}\right)^2\right)} \tag{13}$$

where ω_i and φ_{ij} are ith natural frequency and jth mode shape of ith freedom degree, respectively, whereas n and n_{d} stand for number of variation modes considered and freedom degrees involved in the objective function, respectively. The superscript ex and gp indicate the experimental values and the results of the finite element model gained from the GA-PSO algorithm, respectively. Furthermore, prior using the mode shapes in the objective function, all of them are normalized to have a unit length [25].

2.6 Stage 6: Apply the GA-PSO Algorithm

In order to obtain the best results, the hybrid algorithm described in the Sect. 3 should be run ten times and the best outcome be selected.

2.7 Stage 7: Show the Damage Scenario Found

Finally, the best vector $\boldsymbol{\beta}$ produced by algorithm is reported as the best answer and scenario found. Amount of the vector has the least value of objective function.

3 The Hybrid Algorithm GA-PSO

Metaheuristics classifications can be done by using different criteria. One of the classifications includes population-based and single-solution-based classification. Generally, the single-solution-based metaheuristics are more exploitation-oriented, while the population-based metaheuristics are more exploration-oriented [26]. Metaheuristics, totally, are used to solve complicated optimization problems in different fields from finance to engineering. They are nature inspired and have random variables. Hence, they do not use Hessian matrix and gradient of objective function. The single-solution-based metaheuristics start with a single initial solution and move toward from it; however, a single-solution-based metaheuristic in the search space is concerned with single-solution, while population-based metaheuristics deal with a set like population. Population-based metaheuristics mainly are divided into two general methods related to Evolutionary Computation (EC) and Swarm Intelligence (SI). While the EC algorithms are derived from Darwin's evolutionary theory and are modified population by recombining and mutation operators, the SIs algorithms produce computational intelligence by exploiting simple analogues of social interaction. It is worth mentioning that genetic algorithm (GA) is the subset of EC and particle swarm optimization (PSO) is SI's.

3.1 Genetic Algorithm

GA is the most famous and applied evolutionary computational methods which has been developed in early 1970s by Holland [27]. GA has many aspects being done in different ways considering problems such as solution representation (chromosomes), selection scheme, the crossover type (recombined algorithm operator), and mutation operators. Crossover is mainly known as the main variation operator which is consisted of several individuals (mainly two) selected by replacing some of their parts with the others. Furthermore, some strategies such as n-point and uniform crossover can do this. Parameter pc, which is the crossover

Table 1 The framework of the GA

GA
1. Generate initial population with random individuals
2. Evaluate the cost of every individual
3. Repeat following steps until a termination condition is met
4. Select parents
5. Crossover (recombine) pairs of parents with probability
6. Apply mutation
7. Evaluate new merged individuals
8. Select individuals for the next generation
9. Go to 2 or termination if termination condition is met

rate, shows the probability of being subject to the crossover for each individual, and its value is usually between 0.6 and 1 [28]. In the selection process, individuals are evaluated considering their cost value and are selected to produce offspring using the objective function of the optimization problem. Some selection schemes include roulette-wheel selection, tournament selection, and ranking selection. More information about comparisons of selection schemes is provided in [29]. After applying crossover, the mutation process is applied to the individuals. Thus, new random variables are provided which prohibits the algorithm from trapping in local optima. The mutation rate is pm, and its value is typically determined considering optimization problem. A complete reference and review of genetic algorithms could be found in [30–32]. The framework of the GA is shown in Table 1.

3.2 Particle Swarm Optimization

Particle swarm optimization (PSO), a metaphor of flocking behavior of birds for solving optimization problems, was presented as a general optimization technique by Kennedy [33]. Lots of autonomous particles are produced in the search space randomly that each particle is a representative of solution and shows a location in the search space by a velocity. A swarm includes N moving particle around D dimensional of the search space. Also, each particle has a memory helping it remember its best previous position. A set of particles in which the particle i is connected topologically to them is known as i's neighborhood. The neighborhood might include all the population or some parts of its subset. To identify other particles for influencing on individuals, various topologies were used. Initializing is done for all individuals in the algorithm randomly. In every exploration, each particle is updated by two values. The first one, $\vec{P}_i$, belongs to the best position that the particle has experienced so far and is called personal best. The second one, $\vec{P}_g$, which the algorithm seeks to find it, is the best position gained in the population called global best. After finding the two best, the particle's position, $\vec{X}_i$, and velocity, $\vec{V}_i$, which show particle's direction and location, respectively, are updated as follows:

$$V_{id}(t+1) = \chi\left(V_{id}(t) + C_1 r_1 \left(P_{id}(t) - X_{id}(t)\right) + C_2 r_2 \left(P_{gd}(t) - X_{id}(t)\right)\right) \quad (14)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (15)$$

where i = 1, 2, …, N and N is the size of the swarm. r_1 and r_2 are random numbers uniformly distributed between 0 and 1, whereas C_1 and C_2 are acceleration coefficients representing the attraction of a particle toward its own success and toward success of its neighbors, respectively. χ is the constriction factor being obtained as follows [34]:

$$\chi = \frac{2}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}} \quad (16)$$

Fig. 3 The framework of the PSO

PSO
1. Generate a swarm with N particles
2. Initialize the position and velocity of every particle randomly on D dimensions in the search space
3. while termination condition is not achieved
4. for i = 1: N
5. Update velocity of every particle using Eq. (14)
6. Update position of every particle using Eq. (15)
7. Evaluate the cost value $f(\vec{X}_i)$
8. if $(f(\vec{X}_i) > f(\vec{P}_i))$
9. then $\vec{P}_i \leftarrow \vec{X}_i$
10. end if
11. if $(f(\vec{X}_i) > f(\vec{P}_g))$
12. then $\vec{P}_g \leftarrow \vec{X}_i$
13. end if
14. end for i
15. end while

Fig. 4 The framework of the GA-PSO

GA-PSO

1. Create initial population with N individuals
2. Initialize the position and velocity of every individual randomly
3. Evaluate the cost of every individual
4. Determine the $\overrightarrow{P_i}$ and $\overrightarrow{P_g}$
5. while termination condition is not met
6. for $i = 1$: *MaxGA*
7. Select parents
8. Crossover pair of parents
9. Determine $\overrightarrow{P_i}$ of the resulting offspring
10. Determine velocity of the resulting offspring
11. Apply mutation
12. Determine velocity and $\overrightarrow{P_i}$ of chromosomes subjected to mutation
13. Merge and sort population
14. Delete extra individuals
15. Update $\overrightarrow{P_i}$ and $\overrightarrow{P_g}$
16. end for i
17. for $j = 1$: *MaxPSO*
18. for $k = 1$: N
19. Update velocity of every particle using Eq. (14)
20. Update position of every particle using Eq. (15)
21. Evaluate the cost value
22. Update the $\overrightarrow{P_i}$ and $\overrightarrow{P_g}$
23. end for k
24. end for j
25. end while

Table 2 The details of GA operators

MaxGA	Chromosome	Selection	Crossover	Mutation
2	Binary	Tournament $n = 3$	Two points, $p_c = 0.85 \sim 0.9$	Jump, $p_m = 0.02$

where φ_1 and φ_1 are considered equal to 2.05 and $\varphi = \varphi_1 + \varphi_2 > 4$.

General structure of the PSO algorithm is shown in Fig. 3. Several survey articles regarding some studies related to the applications of PSO could be found in [35,36].

3.3 GA-PSO

The outline of hybrid algorithm GA-PSO is summarized in Fig. 4. The algorithm procedure is divided into three sections: The first one is called initializing which includes the lines first to four; the second one, which consists of GA operators, includes the lines 6 to 16; and the third one, operating PSO, includes the lines 17 to 24. According to these three sections, the operational procedure of the hybrid algorithm could be explained as the following subsections.

3.3.1 Section 1

It is possible to consider population members either as individuals in the GA or particles in the PSO. In this study, they are considered as individuals. These individuals not only have properties of the individuals in GA, but have particles properties in PSO as well. Therefore, each individual has velocity in addition to the position which should be initialized randomly. Then cost values related to each particle (individual) are evaluated regarding to the objective function researched. To determine the personal and global best in this category, the position and cost values of each individual's personal best

Table 3 The details of PSO

MaxPSO	Velocity range	Variable range	C_1, C_2
3	[−0.1, 0.1]	[0, 1]	$\chi\varphi_1, \chi\varphi_2$

are equalized with the same values of initial position and cost evaluated, respectively. Consequently, the cost value of global best equals cost value of the best (least) of personal best.

The main loop of the hybrid algorithm shown in Fig. 4 will work continually until a termination condition, which is the number of assumed loops for the algorithm, is satisfied. Therefore, the steps related to categorys 2 and 3 are performed for every assumed loop.

3.3.2 Section 2

In this part, the GA operators are implemented on their individuals. The MaxGA parameter is the number of iteration of inner GA loop. In other words, this parameter is the iteration number of implementation of the GA on individuals. Considering the type of assumed scheme selection, a pair of parents is selected among population. Then, crossover should be applied to parents, and offspring are produced. Since the values of the bests and velocity are specific to PSO, they cannot be determined in a specific way in this section. To determine offspring's best personal best cost, the value of the personal best of the parent, having the lesser cost, is given to the offspring. On the other hand, a random number between 0 and 1 is produced to determine the offspring's velocity. If the number is under 0.5, the velocity of the first and second parent is given to the first and second offspring, respectively. Otherwise, the velocity of first and second parent is given to the second and first offspring, respectively. Considering the assumed mutation rate, mutation is applied to the main population (not the offspring produced by crossover). Also, the velocity and personal best of the chromosomes subjected to mutation are the same as velocity and personal best of the main chromosomes that are not mutated. The population produced by the crossover, mutation, and the initial main population are merged and then sorted. Therefore, extra individuals are deleted. Then, the values of personal and global best are updated for the individuals. All the processes related to this part are iterated MaxGA times. Ultimately, the final population produced by this part with its all properties is inserted to the third section.

3.3.3 Section 3

For all individuals inserted into this part, the following processes are repeated MaxPSO times. First of all, the values of velocity and position are updated using Eqs. (14) and (15). It is worth mentioning that after updating the two values, the accuracy of getting the boundary conditions should be controlled. The determined velocity should be kept in the range of $-V_{\max}$ to $+V_{\max}$ [37], and the updated position should be in defined range for the variables. Second, the evaluated cost values and the personal and global best should be updated. Finally, the cost value of obtained global best of every loop of the main hybrid algorithm is reported, and its position is identified as the damage detection problem solution.

The details of part 2 (GA) and 3 (PSO) are shown in Tables 2 and 3, respectively. In addition, the GA's and PSO's details which are common with hybrid algorithm's details are equal.

4 Numerical examples

In this section, several numerical examples composed of three different plates including an L-shaped two-clamped supported plate, one quarter of a plate with a hole, and a rectangular two-clamped plate are simulated with several scenarios with and without noise. To investigate the performance of the proposed hybrid algorithm, the results of it are compared with GA and PSO algorithms considering the same tested scenarios. The number of considered modes in the objective function is one of the main inputs for the problem. Although existence of many numbers helps the algorithm easily converges the right state of damage, it increases the rate of program running and results in more time to the algorithm. Therefore, the number of considered modes in the objective function should be selected in a way that induces appropriate balance for the algorithm. The number of assumed modes in the plates is stated in each example. It is worth mentioning that the number of freedom degrees in each mode of examples equals to the total number of freedom degrees involved in the modes. In all of the tables in this article, '*' and '†' symbols denote the noise-free and noisy conditions, respectively. The results are reported to three decimal places, and the values less than 0.01 are considered equal to zero. Furthermore, number of successful runs (on ten) for different scenarios of every example is reported in the tables, and the misidentified elements found by the proposed approach are underlined for a better understanding of the results. In optimization problems, the time of program running is an important factor. Indeed, the less time for algorithm to get an appropriate answer, the more effective the method and the algorithm are. In order to draw a logical comparison between the results of the mentioned algorithms, time is considered as a base factor in selecting inputs such as iterations number and population size in algorithm. Moreover, in order to obtain an equal running time for the two hybrid algorithms, the size of population and the number of iterations for GA and PSO algorithms are selected in three different examples.

Thus, the population size for the three algorithms GA-PSO, GA, and PSO is assumed 250, 800, and 850, respectively, for the first plate. For the second and third plates, the population size is supposed to be 350, 1000, and 1100, respectively. Also, in all examples, the number of iterations for GA-PSO, GA, and PSO algorithms is selected as 800, 1600, and 1600, respectively.

The three examples are an L-shaped two-clamped supported plate, one quarter of a plate with a hole, and a rectangular two-clamped supported plate which have 32, 36, and 40 CST elements and 27, 28, and 30 nodes, respectively (Figs. 5, 6, 7). In this study, the physical properties of the plates are given in Table 4.

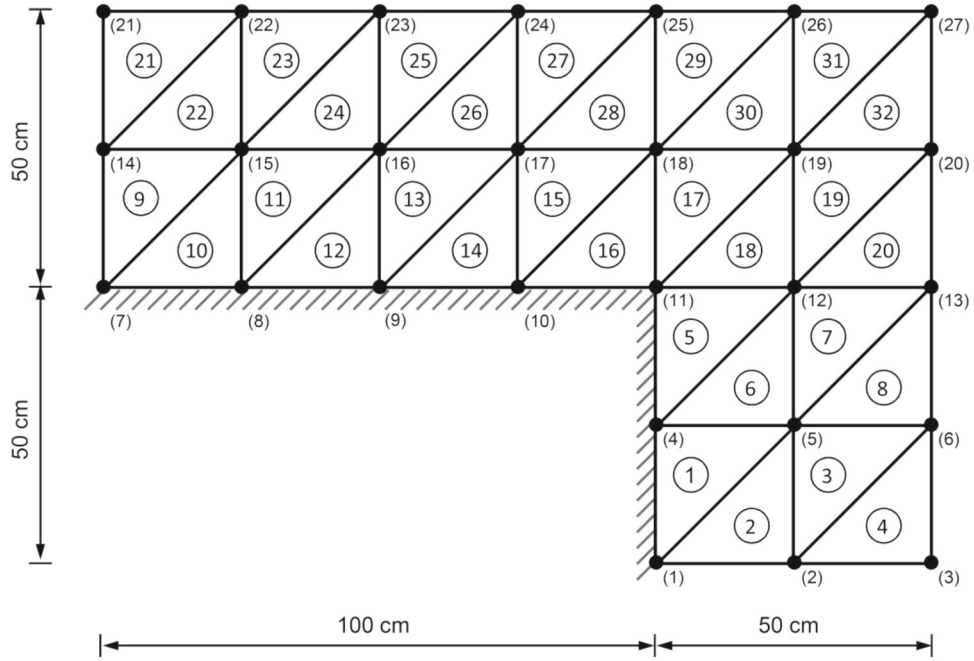

Fig. 5 An L-shaped two-clamped supported plate

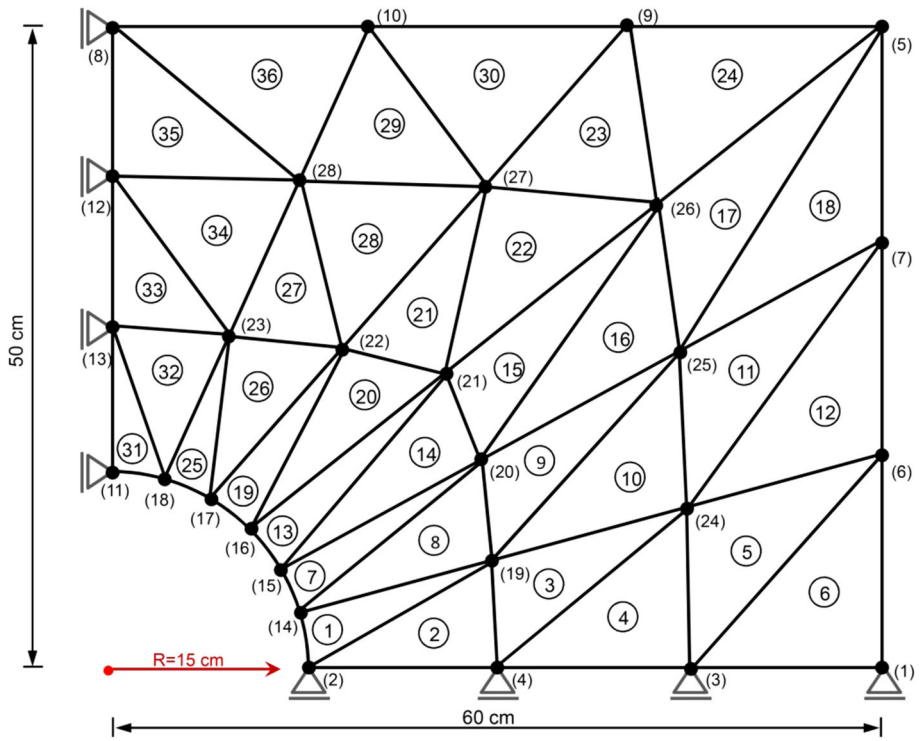

Fig. 6 One quarter of a plate with a hole

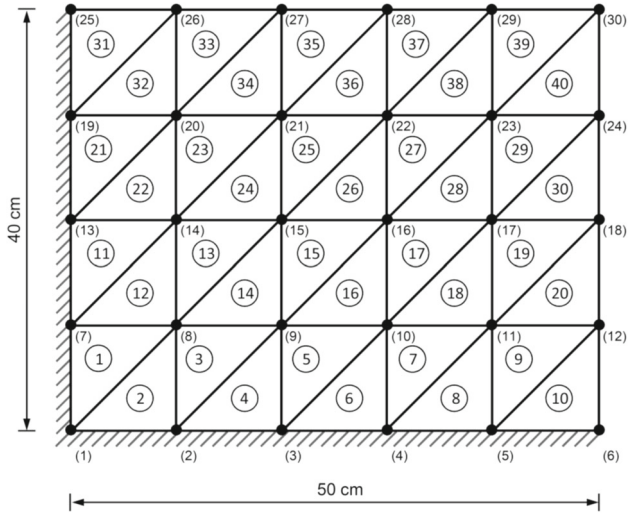

Fig. 7 A rectangular two-clamped supported plate

4.1 An L-Shaped Two-Clamped Supported Plate

The five first modes are considered, and the four following damage scenarios are assumed:

Scenario 1: 20% damage in element 16
Scenario 2: 10% damage in element 9, 12% damage in element 32
Scenario 3: 10% damage in element 9, 7% damage in element 13, and 12% damage in element 32
Scenario 4: 25% damage in element 15, 22% damage in element 19, 14% damage in element 20, and 12% damage in element 32

Table 5 shows the results of applying the hybrid algorithm to the plate for the four damage scenarios.

The evolutionary process of damage severities of the damaged element corresponding to scenario 1 for noisy condition is shown in Fig. 8.

According to Table 5, high number of damaged elements results in less number of successful runs.

When noise is added to the problem, some errors are created in the experimental modal data. However, there is no corresponding real $\boldsymbol{\beta}$ vector data. Thereupon, the algorithm seeks the $\boldsymbol{\beta}$ which has the least divergence from the model data. Consequently, the found $\boldsymbol{\beta}$ can (or cannot) be the same experimental assumed scenario. Although the found $\boldsymbol{\beta}$ of a scenario in noisy condition is equal to its assumed experimental one, the value of the objective function is not zero in the last iteration. In the case noisy condition, this error in objective function is due to the noise.

Table 4 physical properties of the plates

Property (unit)	Value
E, elasticity modulus (GPa)	210
ρ, mass density (kg/m^3)	7850
ν, Poisson's ratio	0.3
t, thickness (m)	0.005

4.2 One Quarter of a Plate With a Hole

The six first modes are considered. In this plate, the four following damage scenarios are presumed:

Scenario 1: 6% damage in element 7
Scenario 2: 15% damage in element 13, 10% damage in element 24
Scenario 3: 5% damage in element 1, 8% damage in element 18, and 10% damage in element 31
Scenario 4: 18% damage in element 1, 13% damage in element 13, 20% damage in element 31, and 10% damage in element 35

Table 6 shows the results of applying the hybrid algorithm to the plate for the four damage scenarios.

The evolutionary processes of damage severities of damaged elements corresponding to the scenario 2 for noisy condition are shown in Fig. 9.

4.3 A Rectangular Two-Clamped Supported Plate

The six first modes are considered, and the four following damage scenarios are assumed:

Scenario 1: 10% damage in element 15
Scenario 2: 8% damage in element 25, 15% damage in element 40
Scenario 3: 6% damage in element 22, 8% damage in element 25, and 12% damage in element 39
Scenario 4: 25% damage in element 8, 26% damage in element 28, 12% damage in element 39, and 15% damage in element 40

Table 7 shows the results of applying the hybrid algorithm to the plate for the four damage scenarios.

The evolutionary processes of damage severities of damaged elements corresponding to scenario 3 for noisy condition are shown in Fig. 10.

Table 5 Results for the L-shaped two-clamped supported plate

Scenario	Damaged element(s)	Damage severity	Runs
1*	16	0.200	9
1†	16	0.199	6
2*	9, 32	0.100, 0.120	8
2†	9, 21, 32	0.102, 0.020, 0.084	6
3*	9, 11, 13, 22, 32	0.100,0.012, 0.057, 0.03, 0.12	5
3†	9, 13, 32	0.122, 0.040, 0.114	5
4*	15, 19, 20, 32	0.249, 0.221, 0.137, 0.119	5
4†	3, 4, 5, 15, 19, 20, 32	0.012, 0.015, 0.020, 0.253, 0.215, 0.145, 0.128	4

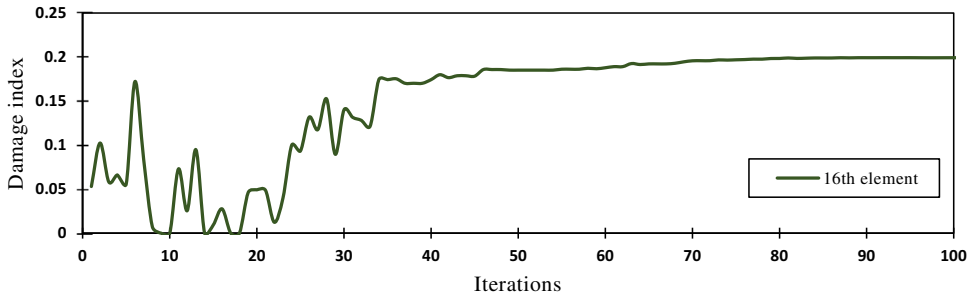

Fig. 8 Evolutionary processes of damage severities for the L-shaped plate corresponding to scenario 1 in 100 first iteration

Table 6 Results for the one quarter of a plate with a hole

Scenario	Damaged element(s)	Damage severity	Runs
1*	7	0.060	9
1†	7	0.056	6
2*	13, 24	0.143, 0.098	8
2†	13, 20, 24	0.111, 0.015, 0.086	7
3*	1, 18, 31	0.049, 0.075, 0.100	6
3†	1, 18, 31	0.049, 0.079, 0.097	6
4*	1, 13, 19, 25, 31, 35, 36	0.156, 0.127, 0.018, 0.012, 0.196, 0.072, 0.015	5
4†	1, 13, 25, 31, 32, 35, 36	0.159, 0.118, 0.021, 0.192, 0.014, 0.073, 0.013	5

4.4 Comparing the Algorithms' Performance

As it was previously mentioned, considering details of the three algorithms, the scenarios related to analyzed plates are tested by PSO and GA for noisy and noise-free conditions. To compare the results of these algorithms and hybrid algorithm, the following equations are used:

$$I_1 = \sum_{i=1}^{m} \left| \frac{\beta_i - \hat{\beta}_i}{\beta_i} \right| \times 100 \tag{17}$$

$$I_2 = \sum_{i=1}^{n-m} \hat{\beta}_{imis} \tag{18}$$

where β_i and $\hat{\beta}_i$ are actual and estimated damage of the ith damaged element using algorithm, respectively. $\hat{\beta}_{imis}$ denotes the estimated damage of the ith undamaged element. m and n represent the number of damaged elements and the number of elements of the structure, respectively.

Considering the details of the Eqs. (17) and (18), I_1 is index of evaluating the percent of error in the damaged elements (elements not zero in the assumed experimental $\boldsymbol{\beta}$ vector) estimated by the algorithm. Since the index is not efficient in demonstrating algorithm's capability, to evaluate undamaged elements (elements in which are assumed experimental $\boldsymbol{\beta}$ vector are zero), the I_2 index is introduced separately which is sum of the values of misidentified elements in each scenario. Hence, it can be said that the nearer these values to zero, the more efficient the algorithm. Tables 8, 9 and 10 show I_1 and I_2 values of the examples as $I_1 + I_2$.

Considering Tables 8, 9 and 10, most of the I_1 and I_2 values corresponding to the hybrid algorithm are less than other values. However, in the scenario 2 corresponding to the first plate and the scenario 1 corresponding to the third plate, I_1

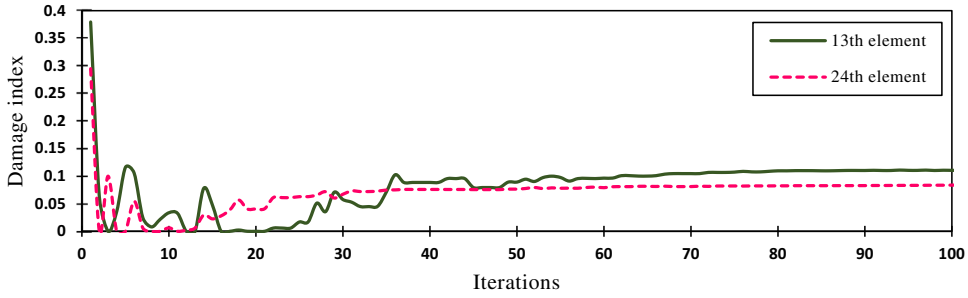

Fig. 9 Evolutionary processes of damage severities for the one quarter of a plate with a hole corresponding to scenario 2 in 100 first iteration

Table 7 Results for the rectangular two-clamped supported plate

Scenario	Damaged element(s)	Damage severity	Runs
1*	15	0.100	8
1†	14, 15	0.015, 0.091	7
2*	25, 40	0.077, 0.146	8
2†	25, 36, 40	0.074, 0.011, 0.147	6
3*	22, 25, 39	0.060, 0.080, 0.120	4
3†	22, 25, 39	0.054, 0.074, 0.116	5
4*	8, 28, 39, 40	0.250, 0.260, 0.120, 0.150	4
4†	8, 28, 33, 39, 40	0.249, 0.248, 0.013, 0.116, 0.135	4

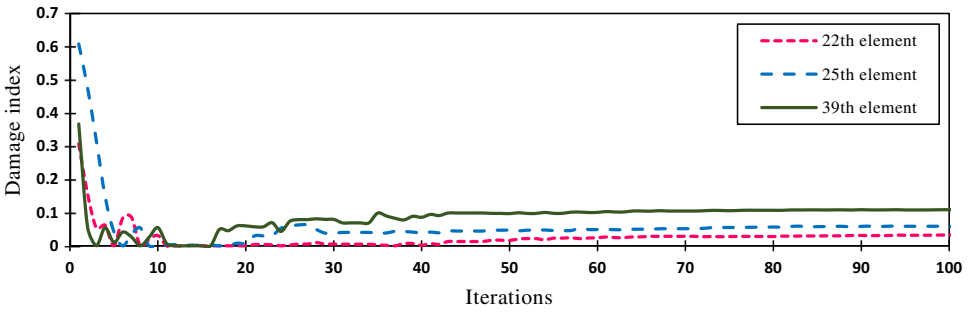

Fig. 10 Evolutionary processes of damage severities for the rectangular plate corresponding to scenario 3 in 100 first iteration

Table 8 Sum error index for the L-shaped two-clamped supported plate

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
GA-PSO*	0.000 + 0.000	0.000 + 0.000	34.279 + 0.044	3.763 + 0.010
GA-PSO†	0.333 + 0.000	31.245 + 0.020	69.976 + 0.009	13.738 + 0.120
PSO*	0.202 + 0.000	7.904 + 0.014	74.739 + 0.071	53.983 + 0.888
PSO†	3.050 + 0.000	17.721 + 0.049	103.641 + 0.329	46.324 + 0.743
GA*	0.064 + 0.000	2.594 + 0.004	42.347 + 0.028	69.948 + 0.353
GA†	12.300 + 0.347	8.798 + 0.157	179.183 + 0.011	98.472 + 0.731

Table 9 Sum error index for the one quarter of a plate with a hole

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
GA-PSO*	0.000 + 0.000	6.977 + 0.013	6.891 + 0.004	45.539 + 0.078
GA-PSO†	5.823 + 0.000	39.974 + 0.025	6.233 + 0.000	52.267 + 0.088
PSO*	0.404 + 0.000	51.548 + 0.094	58.305 + 0.028	110.611 + 0.194
PSO†	7.482 + 0.009	86.707 + 0.738	101.410 + 0.486	169.076 + 0.303
GA*	0.237 + 0.000	43.571 + 0.075	39.956 + 0.021	102.731 + 0.133
GA†	20.252 + 0.063	156.937 + 0.430	70.485 + 0.202	162.881 + 0.112

Table 10 Sum error index for the rectangular two-clamped supported plate

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
GA-PSO*	0.000 + 0.000	6.093 + 0.007	0.001 + 0.000	0.000 + 0.000
GA-PSO†	9.275 + 0.026	8.824 + 0.026	20.678 + 0.010	18.738 + 0.000
PSO*	1.505 + 0.002	27.764 + 0.034	183.247 + 0.144	16.680 + 0.047
PSO†	17.686 + 0.055	43.647 + 0.001	162.137 + 0.623	25.373 + 0.365
GA*	0.411 + 0.000	32.713 + 0.056	172.875 + 0.161	22.968 + 0.045
GA†	1.411 + 0.284	59.063 + 0.103	185.529 + 0.336	120.147 + 0.095

values of GA and PSO are better than GA-PSO. Also in the scenario 3 corresponding to the first plate and the scenario 2 corresponding to the third plate, I_2 values of GA and PSO, respectively, are better than GA-PSO. For all three plates, I_1 maximum values in noisy condition belong to the scenarios 3 and 4 and GA; also in noise-free condition belong to the scenarios 3 and 4 and PSO. Thus, it is concluded that GA and PSO have the most errors in the damaged elements of multiple scenarios for noisy and noise-free condition, respectively. In general, the results concerned hybrid algorithm are better than the other two algorithms. Algorithms provide different solutions, depending on the type of scenario and structure. Since existence of noise in damage detection problem is very important issue and solutions of noisy condition are better by hybrid algorithm, it can be concluded that the algorithm is more efficient in noisy condition setting.

5 Conclusion

In this paper, in addition to introducing a new approach for damage detection of thin plates, a hybrid algorithm, GA-PSO, has been used to solve the inverse optimization damage detection problem in these structures. To formulate the objective function, dynamic methods based on modal are used. The methods are based on the studying of the changes in the structure's properties. Therefore, their underlying theory is based on the fact that damage makes some changes in the dynamic parameters of the structure. Through the modal properties of the structures, natural frequencies and mode shapes are chosen to be used in the objective function. The effectiveness of the algorithm and the approach is evaluated by simulating the three different thin plate structures with different scenarios in two conditions of with and without noise. Moreover, considering the importance of the time in the optimization problems, the effectiveness of the hybrid algorithm is determined, especially in added-noise condition, by comparing the results of the presented algorithm with two GA and PSO algorithms spending almost equal times for running. Among all the tested damage scenarios by hybrid algorithm, few misidentified elements were found with damage severity less than 0.05 and there was no damaged element not found.

References

1. Doebling, S.W.; Farrar, C.R.; Prime, M.B.; Shevitz, D.W.: Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review. In: Los Alamos National Lab, NM (United States) (1996)
2. Lee, E.-T.; Eun, H.-C.: Damage detection of damaged beam by constrained displacement curvature. J. Mech. Sci. Technol. **22**(6), 1111–1120 (2008)
3. Carden, E.P.; Fanning, P.: Vibration based condition monitoring: a review. Struct. Health Monit. **3**(4), 355–377 (2004)
4. Doebling, S.W.; Farrar, C.R.; Prime, M.B.: A summary review of vibration-based damage identification methods. Shock Vib. Dig. **30**(2), 91–105 (1998)
5. Fan, W.; Qiao, P.: Vibration-based damage identification methods: a review and comparative study. Struct. Health Monit. **10**(1), 83–111 (2011)
6. Rytter, A.: Vibration Based Inspection of Civil Engineering Structures, 1993. Aalborg University, Denmark (1993)
7. Bagheri, A.; Amiri, G.G.; Razzaghi, S.S.: Vibration-based damage identification of plate structures via curvelet transform. J. Sound Vib. **327**(3), 593–603 (2009)
8. Wang, Q.; Deng, X.: Damage detection with spatial wavelets. Int. J. Solids Struct. **36**(23), 3443–3468 (1999)
9. Yan, Y.; Yam, L.: Online detection of crack damage in composite plates using embedded piezoelectric actuators/sensors and wavelet analysis. Compos. Struct. **58**(1), 29–38 (2002)
10. Rucka, M.; Wilde, K.: Application of continuous wavelet transform in vibration based damage detection method for beams and plates. J. Sound Vib. **297**(3), 536–550 (2006)
11. Kim, M.; Kim, E.; An, Y.; Park, H.; Sohn, H.: Reference-free impedance-based crack detection in plates. J. Sound Vib. **330**(24), 5949–5962 (2011)
12. Fan, W.; Qiao, P.: A 2-D continuous wavelet transform of mode shape data for damage detection of plate structures. Int. J. Solids Struct. **46**(25), 4379–4395 (2009)
13. Navabian, N.; Bozorgnasab, M.; Taghipour, R.; Yazdanpanah, O.: Damage identification in plate-like structure using mode shape derivatives. Arch. Appl. Mech. **86**(5), 819–830 (2015)
14. Xiang, J.-W.; Matsumoto, T.; Long, J.-Q.; Ma, G.: Identification of damage locations based on operating deflection shape. Nondestruct. Test. Eval. **28**(2), 166–180 (2013)
15. Song, W.; Dyke, S.; Yun, G.; Harmon, T.: Improved damage localization and quantification using subset selection. J. Eng. Mech. **135**(6), 548–560 (2009)
16. Nicknam, A.; Hosseini, M.: Structural damage localization and evaluation based on modal data via a new evolutionary algorithm. Arch. Appl. Mech. **82**(2), 191–203 (2012)
17. Masoumi, M.; Jamshidi, E.: Damage diagnosis in steel structures with different noise levels via optimization algorithms. Int. J. Steel Struct. **15**(3), 557–565 (2015)
18. Mukhopadhyay, T.; Dey, T.K.; Chowdhury, R.; Chakrabarti, A.: Structural damage identification using response surface-based

multi-objective optimization: a comparative study. Arab. J. Sci. Eng. **40**(4), 1027–1044 (2015)
19. Xiang, J.; Liang, M.: A two-step approach to multi-damage detection for plate structures. Eng. Fract. Mech. **91**, 73–86 (2012)
20. Xiang, J.; Zhong, Y.; Chen, X.; He, Z.: Crack detection in a shaft by combination of wavelet-based elements and genetic algorithm. Int. J. Solids Struct. **45**(17), 4782–4795 (2008)
21. Jena, P.K.; Parhi, D.R.: A modified particle swarm optimization technique for crack detection in Cantilever Beams. Arab. J. Sci. Eng. **40**(11), 3263–3272 (2015)
22. Zare Hosseinzadeh, A.; Ghodrati Amiri, G.; Koo, K.-Y.: Optimization-based method for structural damage localization and quantification by means of static displacements computed by flexibility matrix. Eng. Optim. **48**(4), 543–561 (2016)
23. Chen, B.; Nagarajaiah, S.: Flexibility-based structural damage identification using Gauss-Newton method. In: The 14th International Symposium on: Smart Structures and Materials and Nondestructive Evaluation and Health Monitoring 2007, pp. 65291L-65291L–65212. International Society for Optics and Photonics
24. Kaveh, A.; Hoseini Vaez, S.R.; Hoseini, P.; Fallah, N.: Detection of damage in truss structures using Simplified Dolphin Echolocation algorithm based on modal data. Smart Struct. Syst. **18**(5), 983–1004 (2016)
25. Kaveh, A.; Zolghadr, A.: An improved CSS for damage detection of truss structures using changes in natural frequencies and mode shapes. Adv. Eng. Softw. **80**, 93–100 (2015)
26. Boussaïd, I.; Lepagnot, J.; Siarry, P.: A survey on optimization metaheuristics. Inf. Sci. **237**, 82–117 (2013)
27. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. University of Michigan Press, Ann Arbor (1975)
28. Bäck, T.; Schwefel, H.-P.: An overview of evolutionary algorithms for parameter optimization. Evol. Comput. **1**(1), 1–23 (1993)
29. Blickle, T.; Thiele, L.: A comparison of selection schemes used in genetic algorithms. In. TIK-Report (1995)
30. Beasley, D.; Bull, D.R.; Martin, R.R.: An overview of genetic algorithms: part 2, research topics. Univ. Comput. **15**(4), 170–181 (1993)
31. Beasley, D.; Martin, R.; Bull, D.: An overview of genetic algorithms: part 1. Fundamentals. Univ. Comput. **15**, 58–58 (1993)
32. Michalawicz, Z.: Genetic algorithms + data structures = evolution programs. Springer, Berlin (1996)
33. Kennedy, J.: Particle swarm optimization. In: Encyclopedia of Machine Learning, pp. 760–766. Springer, Berlin (2011)
34. Clerc, M.; Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58–73 (2002)
35. Castillo, O.; Melin, P.: Optimization of type-2 fuzzy systems based on bio-inspired methods: a concise review. Inf. Sci. **205**, 1–19 (2012)
36. Thangaraj, R.; Pant, M.; Abraham, A.; Bouvry, P.: Particle swarm optimization: hybridization perspectives and experimental illustrations. Appl. Math. Comput. **217**(12), 5208–5226 (2011)
37. Eberhart, R.; Simpson, P.; Dobbins, R.: Computational Intelligence PC Tools. Academic Press Professional, Inc, Cambridge (1996)