Smart Structures and Systems, *Vol. 15, No. 5 (2015) 1253-1270*
DOI: http://dx.doi.org/10.12989/sss.2015.15.5.1253

Damage detection based on MCSS and PSO using modal data

Ali Kaveh[*] and Mohsen Maniat

Centre of Excellence for Fundamental Studies in Structural Engineering, Department of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran

(Received November 20, 2013, Revised March 10, 2014, Accepted April 28, 2014)

Abstract. In this paper Magnetic Charged System Search (MCSS) and Particle Swarm Optimization (PSO) are applied to the problem of damage detection using frequencies and mode shapes of the structures. The objective is to identify the location and extent of multi-damage in structures. Both natural frequencies and mode shapes are used to form the required objective function. To moderate the effect of noise on measured data, a penalty approach is applied. A variety of numerical examples including two beams and two trusses are considered. A comparison between the PSO and MCSS is conducted to show the efficiency of the MCSS in finding the global optimum. The results show that the present methodology can reliably identify damage scenarios using noisy measurements and incomplete data.

Keywords: damage detection; natural frequencies; mode shapes; beams; trusses; magnetic charged system search; particle swarm optimization

1. Introduction

Normally In recent years, there has been a considerable demand for more precise techniques to identify damages in structures. It is well-known that damages may considerably change the behavior of the structures. While visual inspection fails to detect damages at the early stages, vibration measurements are sensitive enough to identify the damages, even if it is located in hidden or internal areas. Damage detection, as determined by changes in the dynamic properties or the response of the structures, is a subject that has received considerable attention in the literature. The basic idea is that modal parameters (notably natural frequencies and mode shapes) are functions of the physical properties of the structure (mass, damping, and stiffness). Therefore, changes in the physical properties will cause changes in modal characteristics of the structure. A review of damage detection methodologies based on dynamic parameters can be found in the work of (Doebling *et al.* 1998, Carden and Fanning 2004, Fan and Qiao 2011).

One group of the methods employs the optimization algorithms for detecting the multiple structural damages. These methods use techniques of finite element model updating and error minimization to obtain a correct set of physical parameters that reproduce the measured data. Many successful applications of damage detection utilizing the metaheuristic algorithms have been reported in the literature. Maity and Tripathy (2005) used a genetic algorithm for the detection of structural damage using changes in natural frequencies. Liszkai and Raich (2005) employed some

[*]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)

advanced genetic algorithm representations for the structural damage identification of beams and frames. Sahoo and Maity (2007) proposed a hybrid neuro-genetic algorithm and considered both natural frequencies and strains as input parameters to solve damage detection problem. Majumdar et al. (2012) presented a method to identify structural damages in truss structures from changes in natural frequencies by using ant colony optimization. Miguel *et al.* (2012) utilized a harmony search algorithm for the detection of damage in structures, under ambient vibration. Villalba and Laier (2012) used a multi-chromosome genetic algorithm to locate and quantify the damages in trusses by using natural frequencies and mode shapes.

Natural frequencies and modal shapes are the most popular parameters utilized in damage identification. Although, natural frequencies are fairly simple to measure, they have some drawbacks. One of the problems about using only natural frequencies is that a significant damage cause very small change in natural frequencies, particularly for larger structures, and these changes may be undetected due to noise in measured data. Another problem is that variations in the mass of the structures may introduce uncertainties in the measured frequency changes. To solve these problems research efforts have focused on using changes in the mode shapes. The interesting characteristic of mode shapes is that these are much more sensitive to local damages in comparison to natural frequencies. However, the use of pure mode shapes also has some difficulties. Firstly, damage is a local phenomenon and may have no influence on the lower mode shapes significantly. Secondly, the number of sensors and their positions may have a significant effect on the damage identification procedure (Kim *et al*. 2003).

To moderate the above mentioned drawbacks of using natural frequencies and mode shapes, a combination of both are used in this paper. Here, particle swarm optimizer (PSO) and magnetic charge system search (MCSS) are utilized to solve the optimization problem associated with the detection of the damage in beam-type structures and trusses with different scenarios and incompetent data.

2. Objective function

The present damage identification methodology consists of solving an optimization problem through an objective function based on dynamic parameters. The damping matrix is neglected even though it reduces the natural frequencies slightly. Small local changes in the mass can cause significant changes in the dynamic properties. Hence, it is assumed that the mass of the structure does not change after it has been damaged (Friswell and Mottershead 2001). Here, damage is considered as a reduction in stiffness of the damaged elements, and the stiffness reduction is modeled by a reduction in the elastic modulus. The objective function of the optimization problem will consist of three terms, Eq. (1). The first part is related to the natural frequencies, the second part is related to the mode shapes, and the third one is a penalty against too many damage sites. Due to measurement noise, the tendency will always be to find damage at many elements (Friswell *et al.* 1998), some of which are not necessarily damaged. Thus, a penalty is introduced to weight against an increased number of damage sites. Kaveh *et al.* (2014) used PSO, ray optimization and harmony search, and Kaveh and Zolghadr (2015) utilized the CSS for damage detection of truss structures.

$$cost = E(1 + \beta * penalty) \tag{1}$$

$$E = E_{\phi} + E_{\omega} \tag{2}$$

$$E_{\phi} = \sum_{j=1}^{r} \left\| \frac{\phi_j^m - \phi_j^a}{\phi_j^m + \phi_j^a} \right\| \tag{3}$$

$$E_{\omega} = \sum_{j=1}^{r} \left(\frac{\left(\omega_j^m - \omega_j^a\right)^2}{\left(\omega_j^m\right)^2} \right) \tag{4}$$

Where ω_j^m and ω_j^a are the jth measured and analytical natural frequency of the damaged structure, respectively. ϕ_j^m and ϕ_j^a are the measured and analytical value of the jth mode shapes, respectively. r is the number of measured modes and β is a penalty factor which is related to the type of structure and the closeness of the measured data and the exact data. Here, *penalty* is the number of damaged elements in the analytical model.

3. Particle swarm optimization

The algorithm contains a number of particles, each particle being a possible solution for the objective function. Particles are initialized randomly in the search space of an objective function. In each iteration, the velocity of each particle is updated by means of their best encountered position and the best position encountered by any particle using the following formula

$$v_i^{k+1} = \rho^k v_i^k + c_1 r_1 \left(P_i^k - X_i^k\right) + c_2 r_2 \left(P_g^k - X_i^k\right) \tag{5}$$

Where P_i^k is the best previous position of the ith particle in the kth iteration, and P_g^k is the best global position among all the particles in the swarm in the kth iteration. r_1 and r_2 are random values, uniformly distributed between zero and one. c_1 and c_2 are the cognitive and social scaling parameters respectively, and ρ^k is the inertia weight used to discount the previous velocity of the particle. The position of each particle is updated in each iteration. This is done by adding the velocity vector to the position vector as

$$X_i^{k+1} = X_i^k + v_i^{k+1} \tag{6}$$

Where X_i^k and v_i^{k+1} represent the current position and velocity vectors of the ith particle, respectively.

The inertia weight ρ^k may be defined to vary linearly from a maximum value ρ_{max} to a minimum value ρ_{max}. Velocity vector v_i is limited to a lower bound v_{low} and an upper bound v_{upper}. Different techniques have been used to set some of the PSO parameters, such as fuzzy systems (Shi and Eberhart 2001), self-adaptation (Parsopoulos and Vrahatis 2007) and deterministic adaptation based on p_{best} and g_{best} (Arumugam *et al.* 2008). Ratnaweera *et al.* (2004) proposed a time-varying acceleration coefficient (TVAC), which reduces the cognitive component and increases the social component of acceleration coefficient with time. A large value of c_1 and a small value of c_2 at the beginning, may improve the exploration ability, and a small value of $c_1$and a large value of c_2 allow the particles converge to the global optimum in the latter part of the optimization. In this paper, a nonlinear function is used to calculate $c_1$and c_2 according to the following equations

$$c_1 = c_{max} - \left[\left(c_{max} - c_{\min}\right) \times \frac{iter}{iter_{\max}} \right]^n \tag{7}$$

$$c_1 = c_{max} + \left[(c_{max} - c_{min}) \times \frac{iter}{iter_{max}}\right]^m \tag{8}$$

Where c_{min} and c_{max} are the lower and upper bound for both c_1 and c_2. Here, *iter* is the current iteration number and $iter_{max}$ is the maximum number of allowable iterations. According to the type of the problem, *n* and *m* can be calculated to get more efficient results. In this paper, *n* and *m* are taken to 1 and 2 respectively, and c_{min} and c_{max} are chosen as 0.5 and 2.5, respectively.

The velocities of all the particles are limited to the range specified by v_{min} and v_{max} which is equal to $\pm\left(\frac{X_{min}-X_{max}}{3}\right)$. The inertia weight presented by

$$\rho = (\rho_{max})^{iter} \tag{9}$$

Where ρ_{max} is the inertia weight in the first iteration, and it is taken as ρ_{max}= 0.96.

4. Magnetic charged system search

The MCSS algorithm is an improved version of the standard CSS (Kaveh *et al.* 2013). The structures of both algorithms are nearly the same. The main difference between these two algorithms is that CSS only considers the electric forces, but MCSS uses both magnetic and electric forces (Kaveh 2014). Each solution for the objective function (X_i), called charged particle (CP), contains electric charge. These particles produce electric fields and apply electric forces on each other. When a CP moves, it creates a magnetic field in the space, which imposes magnetic forces on the other CPs. For computing magnetic fields, it has been assumed that CPs move in virtual straight wires with radius of R. Thus, the path of movement of each particle consists of straight wires. These straight wires change their directions by each movement of the CPs, but during the movement, each wire remains straight, Fig. 1. When a CP starts a new movement, the direction of its movement may differ from its previous one, so the direction of the wire which includes the CP during its movement also changes. According to magnetic law, a conducting wire carrying electric current can create magnetic fields in the space. Now virtual wires contain CPs that move on them. By each movement of the CPs, their charges are altered, so during the movement the magnitude of the charge is not constant. This movement of CPs can be comprehended as an electric current in the virtual wire. The current of a wire is the rate at which charge flows through one specified cross-section of the wire. If Δq is the amount of charge that passes through this area in a time interval Δt, the average current I_{avg} will be equal to the charge that passes through the cross-section per unit time

$$I_{avg} = \frac{\Delta q}{\Delta t} \tag{10}$$

Since the time intervals of each movement are set to unity, the average current will be equal to the variation of the charge. For computing the variation of the charges, we consider the start and the end points of the movement of the CPs. Considering these assumptions, Eq. (10) can be written as

$$\left(I_{avg}\right)_{ik} = q_i^k - q_i^{k-1} \tag{11}$$

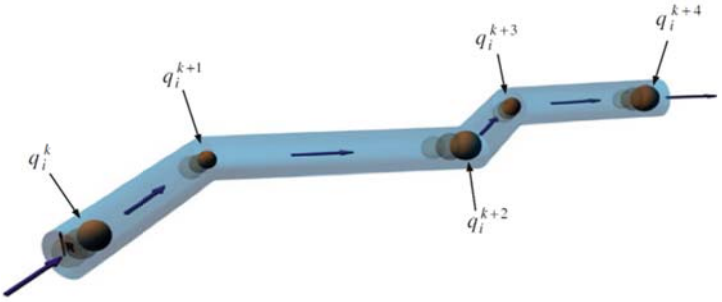


Fig. 1 A schematic view of the virtual wire (Kaveh *et al.* 2013)

Where $\left(I_{avg}\right)_{ik}$ is the average current in the *i*th wire, where the *i*th CP performs its *k*th movement (iteration) in it, and q_i^{k-1} and q_i^k are the charges of the *i*th CP at the start and end of its *k*th movement, respectively. The charges of the CPs are defined by Eq. (12). This expression for computing electric charges results in values between 0 to 1. This is due to normalization of the objective function of each CP. Therefore, the charges of the worst CP and the best CP are always zero and unity, respectively. A normalized value for computing the electric current is proposed as

$$q_i = \frac{fit(i)-fit_{worst}}{fit_{best}-fit_{worst}} \tag{12}$$

$$\left(I_{avg}\right)_{ik} = \frac{q_i^k-q_i^{k-1}}{q_i^k+\varepsilon} \tag{13}$$

Where $q_i^k$and $q_i^{k-1}$are the charge of the *i*th CP at the start of the *k*th and (*k* – 1)th iterations, respectively, and ε is a small positive value to prevent singularity.

The separation distance r_{ij} between two charged particles is defined as follows

$$r_{ij} = \frac{X_i-X_j}{\frac{\left(X_i+X_j\right)}{2}-X_{best}+\varepsilon} \tag{14}$$

Where X_i and X_i-X_j are the positions of the *i*th and *j*th CPs, X_{best} is the position of the best current CP, and ε is a small positive number to avoid singularities.

In the expression for computing the magnetic force, we should consider the velocity of the movement of CPs. In this case, due to the movements of both CPs (CP in the virtual wire and CP in the space), the relative velocity, v_{rel}, is considered as follows which is the modified version of the one in MCSS for better performance of the algorithm

$$v_{rel} = V_i - V_j \tag{15}$$

In this algorithm, each CP is considered as a charged sphere with radius *a*, which has a uniform volume charge density. Here, the magnitude of *a* is set to 0.8. By considering these assumptions, the electric force ($F_{E,j}$) and the magnetic force ($F_{B,j}$) exerted on the *j*th CP can be expressed as

$$F_{E,j} = q_j \sum_{i,i\neq j}\left(\frac{q_i}{a^3} r_{ij}\cdot z_1 + \frac{q_i}{{r_{ij}}^2} z_2\right) p_{ij}\left(X_i-X_j\right), \begin{cases} z_1 = 1, z_2 = 0 \Leftrightarrow r_{ij} < a \\ z_1 = 0, z_2 = 1 \Leftrightarrow r_{ij} \geq a \end{cases} \tag{16}$$

$$F_{B,j} = q_j \sum_{i,i\neq j} \left(\frac{I_i}{R^2} r_{ij} \cdot z_1 + \frac{I_i}{r_{ij}} z_2\right) pm_{ij}(V_i - V_j), \begin{cases} z_1 = 1, z_2 = 0 \Leftrightarrow r_{ij} < R \\ z_1 = 0, z_2 = 1 \Leftrightarrow r_{ij} \geq R \end{cases} \tag{17}$$

Where q_i is the charge of the *i*th CP, *R* is the radius of the virtual wires, *I*$_i$ is the average electric current in each wire, and pm_{ij} is the probability of the magnetic influence (attracting or repelling) of the *i*th wire (CP) on the *j*th CP. This term can be computed by the following expression

$$pm_{ij} = \begin{cases} 1 & \Leftrightarrow \quad fit(j) > fit(i) \\ 0 & \Leftrightarrow \quad else. \end{cases} \tag{18}$$

Where $fit(i)$ and $fit(j)$ are the objective values of the *i*th and *j*th CP, respectively. This probability determines that only a good CP can affect a bad CP by the magnetic force.

Both magnetic and electric forces should be computed and superposed. The Lorentz force (total force) will be expressed as

$$F_j = \rho_{r,j} \times F_{E,j} + F_{B,j} \tag{19}$$

Where F_j is the resultant Lorentz force (total force) acting on the *j*th CP. $\rho_{r,j}$ is the probability that an electrical force is a repelling force, and it is defined as

$$\rho_{r,j} = \begin{cases} 1 & \Leftrightarrow \quad rand > \kappa(1 - iter/iter_{max}) \\ -1 & \Leftrightarrow \quad else. \end{cases} \tag{20}$$

Where *rand* is a random number generated based on a uniform distribution, *iter* is the current number of iterations, *iter*$_{max}$ is the maximum number of iterations and κ is constant parameter that influences in diversification of the searching process. In this paper, κ is set to 0.1.

According to the determined forces, each CP moves to its new position and attains a velocity as

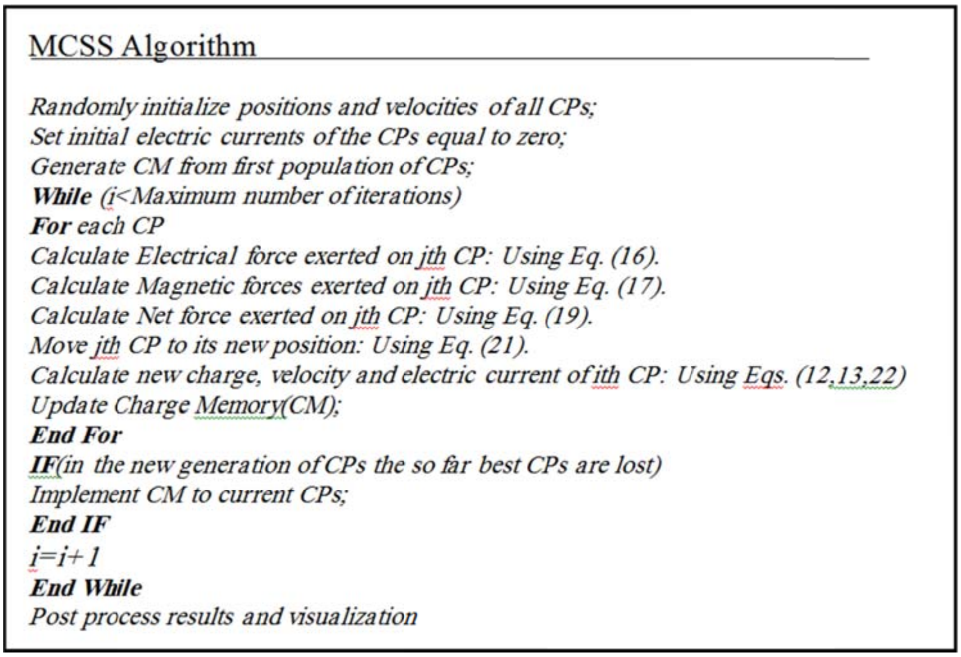
$$X_{j,new} = rand_{j1} \cdot k_a \cdot \frac{F_j}{m_j} \cdot \Delta t^2 + rand_{j2} \cdot k_v \cdot V_{j,old} \cdot \Delta t + X_{j,old} \tag{21}$$

$$V_{j,new=}\frac{X_{j,new} - X_{j,old}}{\Delta t} \tag{22}$$

Where *rand*$_{j1}$ and *rand*$_{j2}$ are two random numbers that are uniformly distributed in the range (0,1). k_a is the acceleration coefficient, k_v is the velocity coefficient, and m_j is the mass of particle which is considered to be equal to q_j . The velocity coefficient controls the influence of the previous velocity of the particles. In other words, this coefficient is related to the exploration ability of the algorithm. The acceleration coefficient affects the force acting on each CP, or it influences the exploitation ability of the algorithm. An efficient optimization algorithm should perform good exploration in early iterations and good exploitation in last iterations. k_a and k_v are expressed as

$$k_a = c_a(1 + \frac{iter}{iter_{max}}) \tag{23}$$

$$k_v = c_v\left(1 - \left(\frac{iter}{iter_{max}}\right)^2\right) \tag{24}$$



```
MCSS Algorithm
Randomly initialize positions and velocities of all CPs;
Set initial electric currents of the CPs equal to zero;
Generate CM from first population of CPs;
While (i<Maximum number of iterations)
For each CP
Calculate Electrical force exerted on jth CP: Using Eq. (16).
Calculate Magnetic forces exerted on jth CP: Using Eq. (17).
Calculate Net force exerted on jth CP: Using Eq. (19).
Move jth CP to its new position: Using Eq. (21).
Calculate new charge, velocity and electric current of ith CP: Using Eqs. (12,13,22)
Update Charge Memory(CM);
End For
IF(in the new generation of CPs the so far best CPs are lost)
Implement CM to current CPs;
End IF
i=i+1
End While
Post process results and visualization
```

Fig. 2 Pseudo-code of the MCSS

Where c_a and c_v are constant parameters. Considering the nonlinear equation for k_v has improved the performance of the original algorithm.

Using this methodology, the new positions of each CP and what it experienced in its last movement can affect the moving process of the subsequent CPs. This method can save a lot of useful information for optimization processes. Based on the explained steps and rules, MCSS algorithm can be summarized as a pseudo-code shown in Fig. 2.

5. Numerical examples

Six numerical examples (two 2D-trusses, a 3D-truss, two beams, and a frame,) with different multi-damage scenarios, consisting of different severity of damages on different elements, are considered here to show the efficiency of the proposed damage identification method in a variety of structures. Modal analysis is conducted by developing a program in MATLAB to get the FE frequencies and mode shapes. Due to the stochastic nature of the metaheuristic algorithms, it is essential to implement the algorithms a specific number of times to detect correct damages. In this paper for each scenario the algorithm has been run ten times and the solution with the lowest cost is selected as the damage scenario searched for. Each run is conducted with five different random noises in order to show that the patterns of the noises do not have significant influence on the solution.

The mode shapes are measured with less accuracy than the natural frequencies. In order to simulate the conditions of a real test, the measured parameters are numerically perturbed by 1% noise for natural frequencies and 3% noise for mode shapes (Villalba and Laier 2012).

5.1 A 25-bar truss

The First example is a statically determinate truss bridge shown in Fig. 3 (Esfandiari *et al.* 2009). This truss has 12 nodes and 25 elements in the finite element model. Area of cross section for all elements is 10cm^2. The modulus of elasticity and the material density are 200 GPa and 7780 kg/m^3, respectively. The first 5 natural frequencies of the structure are used to form the objective function. Three damage scenarios are considered randomly in different elements for the simulated truss. Figs. 4-6 represent the damage states found by the algorithms with the actual damage states in different scenarios.

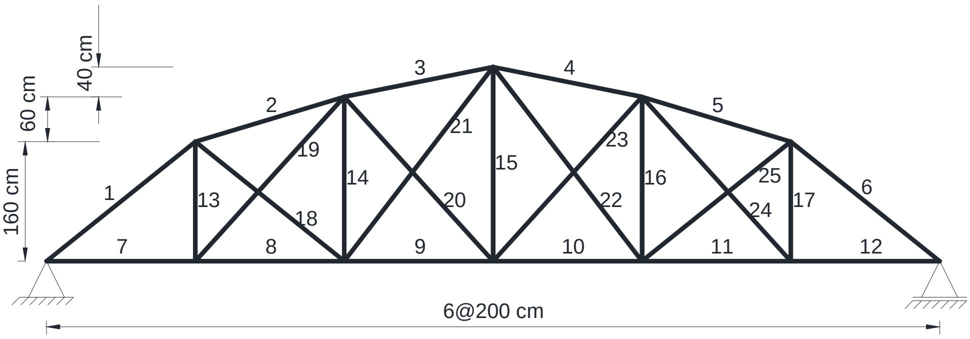


Fig. 3 A truss with 25 elements

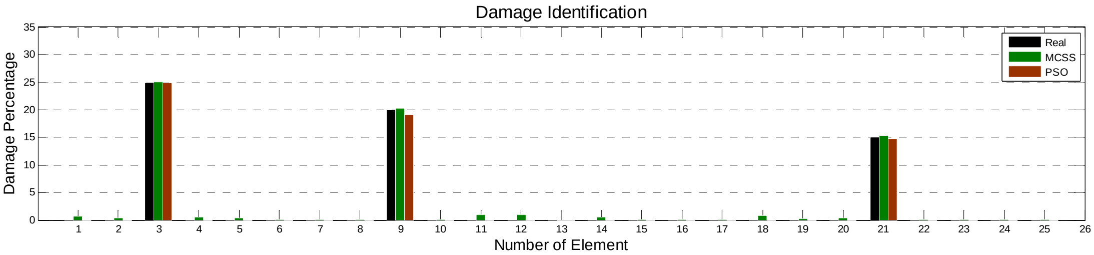


Fig. 4 Damage detection results of the algorithms for the 25-bar truss (scenario I)

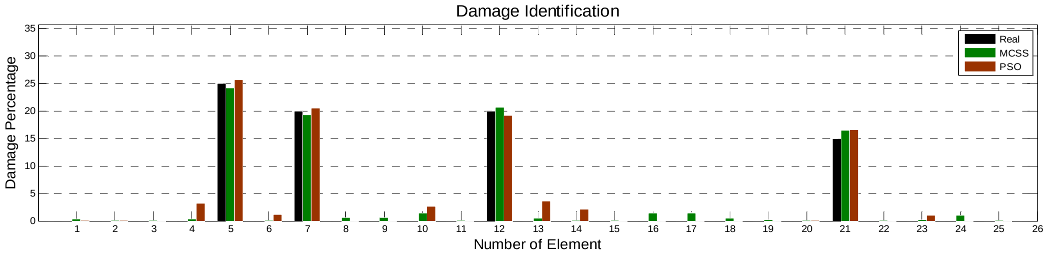


Fig. 5 Damage detection results of the algorithms for the 25-bar truss (scenario II).

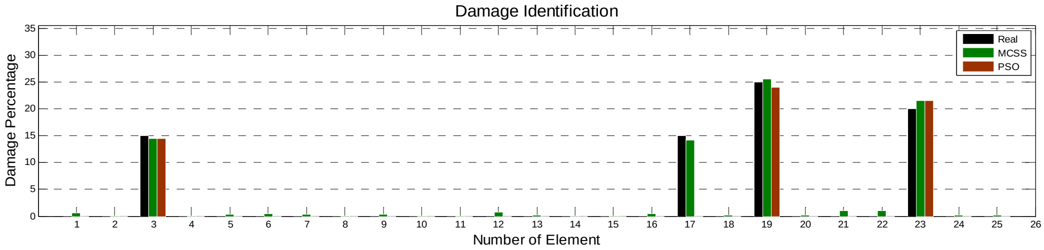


Fig. 6 Damage detection results of the algorithms for the 25-bar truss (scenario III)

In the first two scenarios both algorithms find the correct scenarios with marginal error. Since in the objective function a penalty function has been used, in the first and third scenarios there is no miss-identifications, and in the second scenario all of the miss-identifications have damage percentage less than 4%. As it can be seen from the third scenario, PSO has failed to identify one of the damaged elements; however, MCSS accurately found the location and severity of all of the damaged elements

5.2 A 27-bar truss

The second example is a statically determinate truss bridge as shown in Fig. 7 (Yang and Liu 2007). This truss has 15 nodes and 27 elements in the finite element model. Area of cross section for all elements is 10 cm^2. The modulus of elasticity and the material density are 200 GPa and 7780 kg/m^3, respectively. The first 5 natural frequencies of the structure are used to form the objective function. Three damage scenarios are considered randomly for the simulated truss. Figs. 8-10 represent the damage states found by the algorithm with the actual damage states in different scenarios.

In all of the scenarios represented here, MCSS successfully found the exact location and severity of the damages, with negligible error which is an inevitable consequence of noise in the data. However, in most of the cases PSO has failed to find the correct scenario.

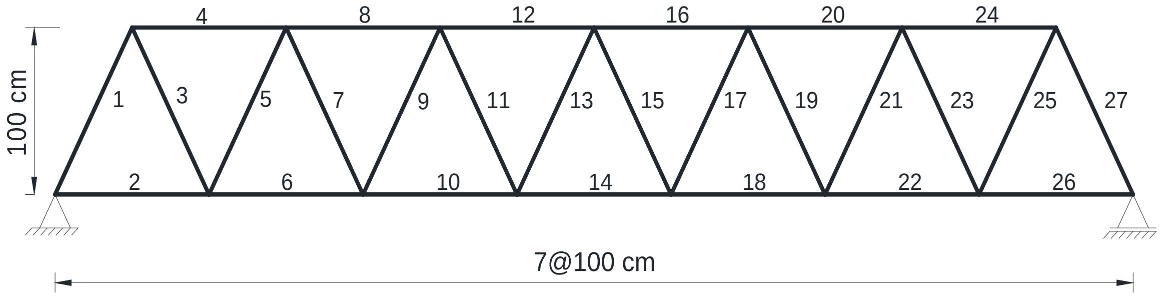


Fig. 7 A truss with 27 elements

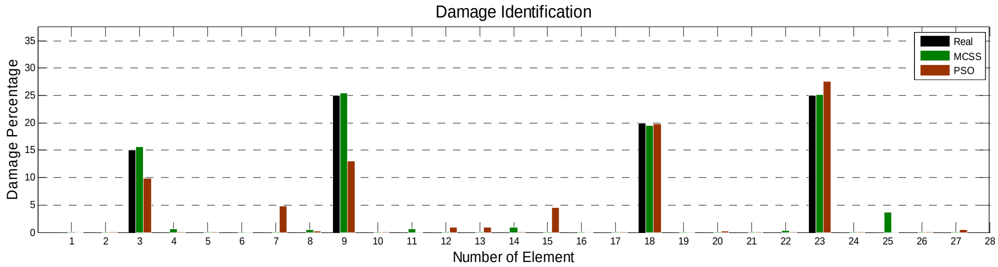


Fig. 8 Damage detection results of the algorithms for the 27-bar truss (Scenario I)

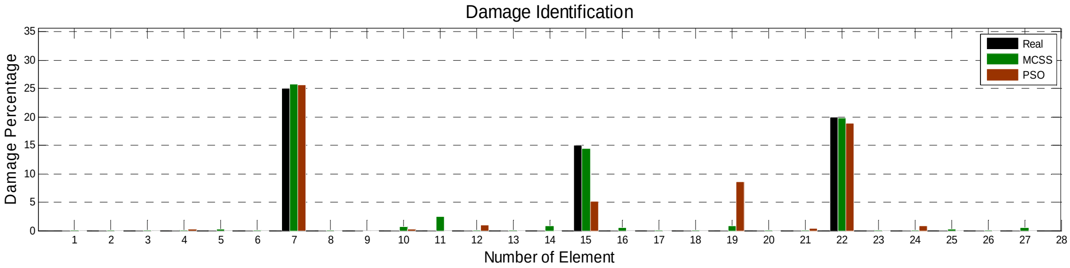


Fig. 9 Damage detection results of the algorithms for the 27-bar truss (Scenario II)

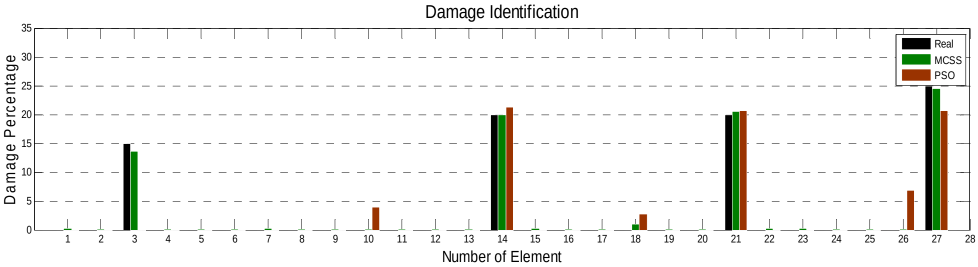


Fig. 10 Damage detection results of the algorithms for the 27-bar truss (Scenario III)

5.3 A 3D-truss

A three dimensional truss is considered as the third example. The geometry, element numbering and material properties are shown in Fig. 11. The first 6 natural frequencies and mode shapes of the structure are used to form the objective function. Three damage scenarios are considered randomly for the simulated truss.

Figs. 12-14 represent the damage states found by both optimization algorithms with the actual damage states in different scenarios. As it can be seen, in the first scenario both algorithms found the correct damage scenarios with no miss-identification. This shows the effectiveness of the proposed objective function. However, in the scenario with three damaged element, PSO has failed to find the exact damage scenarios. In comparison with PSO, the MCSS performed significantly better, and in all of the scenarios it found location and severity of the damages accurately.

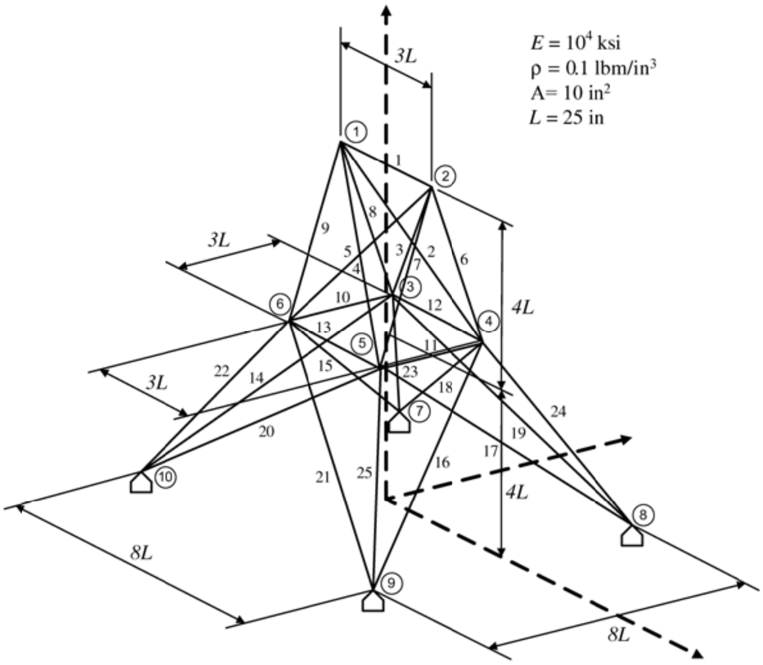


Fig. 11 3D-truss with 25 elements

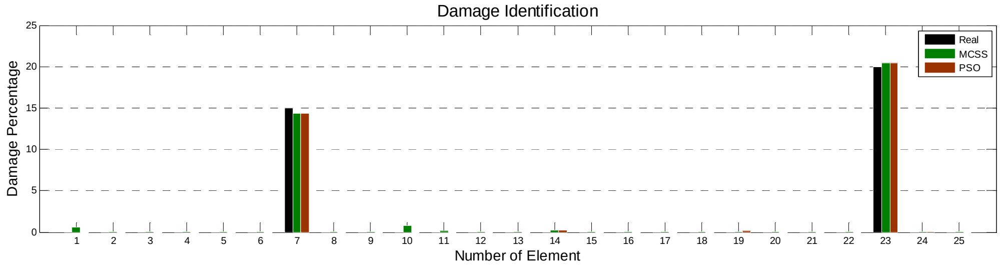


Fig. 12 Damage detection results of the algorithms for the 3D-truss (Scenario I)

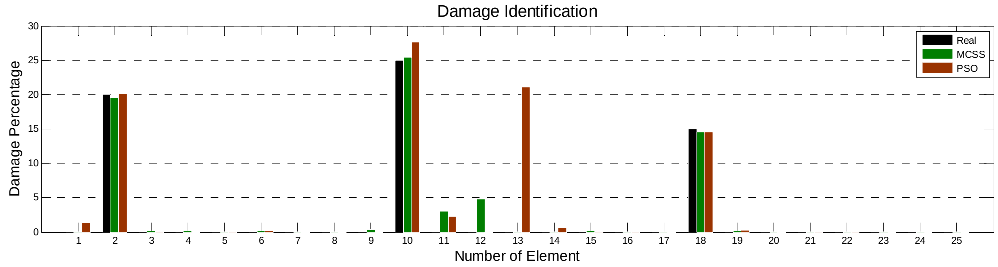


Fig. 13 Damage detection results of the algorithms for the 3D-truss (Scenario II)

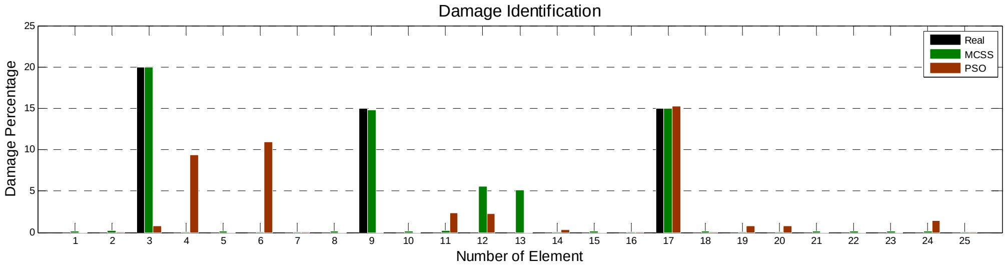


Fig. 14 Damage detection results of the algorithms for the 3D-truss (Scenario III)

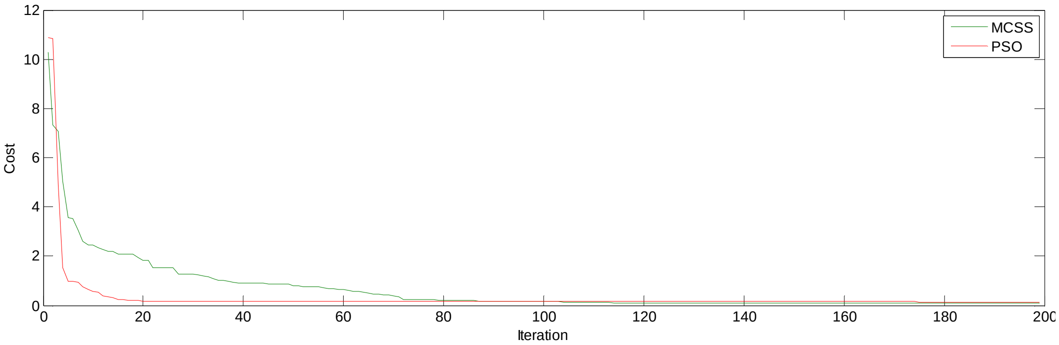


Fig. 15 The convergence of the Scenario III for the 3D-Truss

The convergences of the last scenario for both algorithms have shown in Fig. 15 for comparison. As it can be seen, PSO has a fast convergence, but in this problem fast convergence lead to wrong scenario. MCSS has a smooth convergence, and in the end the value of cost function related to MCSS is less than PSO (0.0919 is the optimum value found by MSCC, and 0.1424 is the optimum value found by PSO).

5.4 A four-span beam

The two-span beam is considered as the fourth example as depicted in Fig. 16. Area of cross section and moment of inertia of the simulated beam are 123.2cm^2 and 22185 cm^4, respectively. The modulus of elasticity and the material density are 210 GPa and 7780 kg/m^3, respectively. The first 5 natural frequencies of the structure are used to form the objective function. Two damage scenarios are considered for the simulated beam near critical elements (elements that are most likely to be damaged due to the maximum bending moment or shear stress are occurring there).

Figs. 17 and 18 represent the damage states found by the algorithm with the actual damage states in different scenarios. In the first scenario both algorithms were successful in detecting damaged elements but in the second scenario PSO failed to detect damage in the first element.

5.5 A four-span beam

The four-span beam depicted in Fig. 19 is considered as the fifth example, in order to show the robustness of the MCSS in large beams with more elements and with different supporting conditions. Area of cross section and moment of inertia of the simulated beam are 123.2 cm^2 and 22185 cm^4, respectively. The modulus of elasticity and the material density are 210 GPa and 7780 kg/m^3, respectively. The first 8 natural frequencies of the structure are used to form the objective function. Four damage scenarios are considered for the simulated beam near the critical elements (elements that are most likely to be damaged due to the maximum bending moment or shear stress are occurring there).

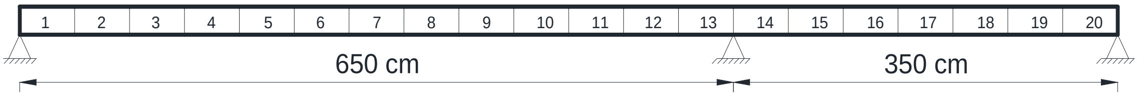


Fig. 16 A two-Span beam modeled with 20 finite elements

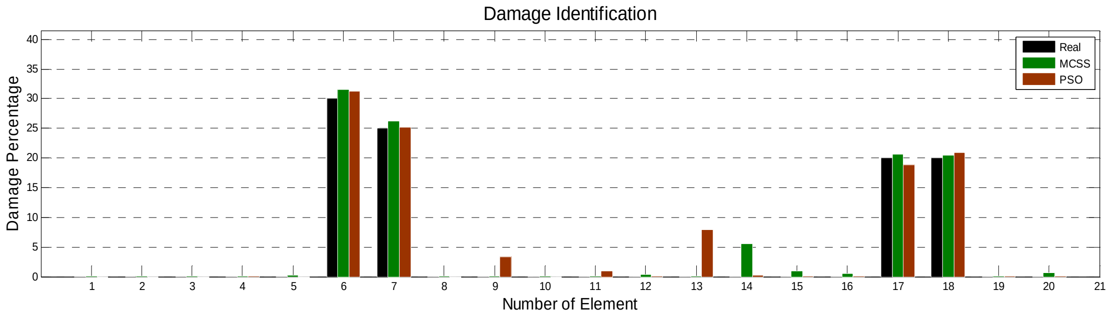


Fig. 17 Damage detection results of the algorithms for two-span beam (scenario I)

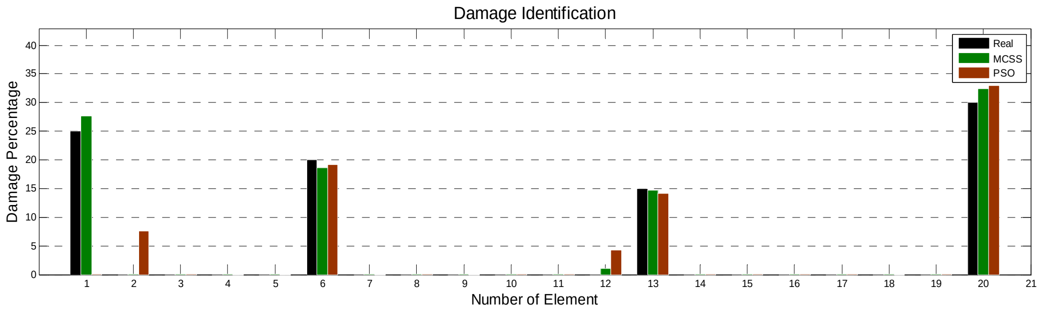


Fig. 18 Damage detection results of the algorithms for two-span beam (scenario II)

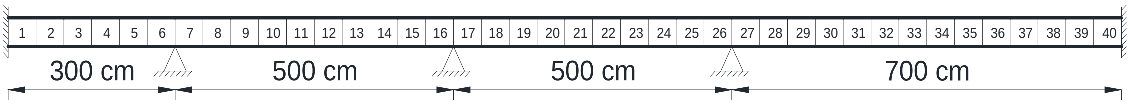


Fig. 19 A four-span beam modeled with 40 finite elements

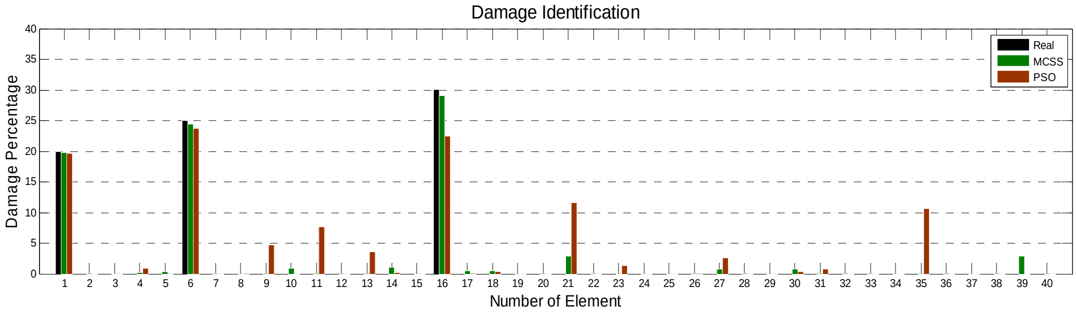


Fig. 20 Damage detection results of the algorithms for the four-span beam (Scenario I)

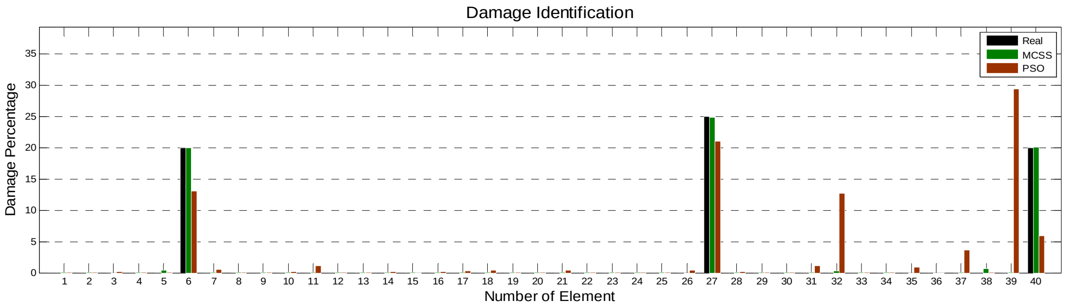


Fig. 21 Damage detection results of the algorithms for the four-span beam (Scenario II)

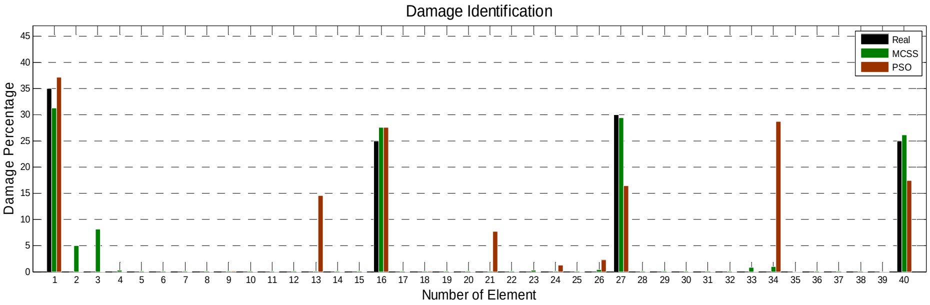


Fig. 22 Damage detection results of the algorithms for the four-span beam (Scenario III).

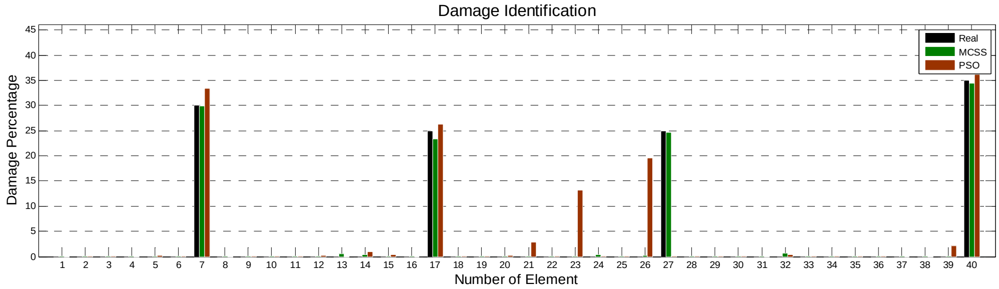


Fig. 23 Damage detection results of the algorithms for the four-span beam (Scenario IV)

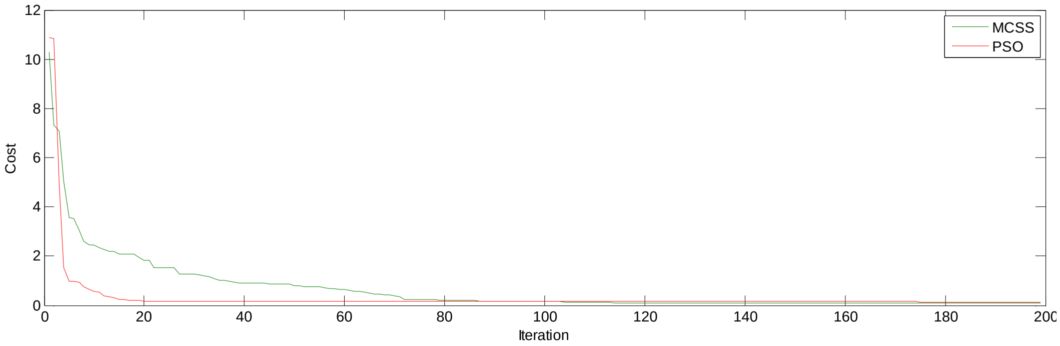


Fig. 24 The convergence of the Scenario IV for the four-span beam

Figs. 20 - 23 represent the damage states found by the algorithms with the actual damage states in different scenarios. Due to the complexity of this example and increase in the number of variables, in most of the scenarios PSO failed to identify the correct damages, but MCSS with insignificant error found the correct damage scenarios.

The convergences of the last scenario for both algorithms have shown in fig. 24 for comparison. The final optimum value found by MCSS and PSO are 0.2151 and 0.2546 respectively.

5.6 A planar frame

The 3-spans and 3-story frame depicted in Fig. 25 is considered as the last example. The sections used for the beams and columns are IPE240 and IPE300, respectively. The modulus of elasticity and the material density are 200 GPa and 7780 kg/m^3, respectively. The first 6 natural frequencies and mode shapes of the structure are used to form the objective function.

Figs. 26 to 28 represent the damage states found by both optimization algorithms with the actual damage states in different scenarios. Although in most of the cases both algorithms found the correct damage scenarios, in one case (Scenario III) PSO was trapped in a local optimum, and identified element number 9 - instead of element number 8 - as the damaged element. However, MCSS found the correct number and location of damaged elements.

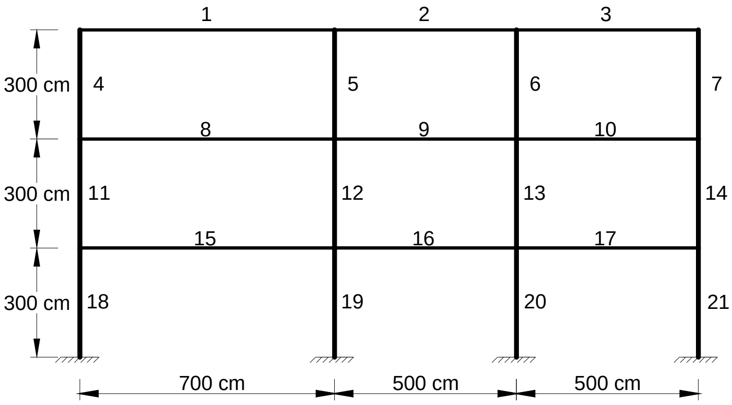


Fig. 25 A three-span three-story frame

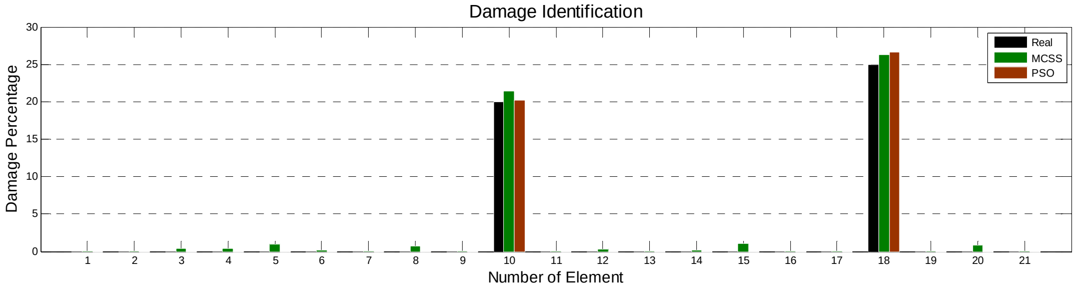


Fig. 26 Damage detection results of the algorithms for the frame (Scenario I)

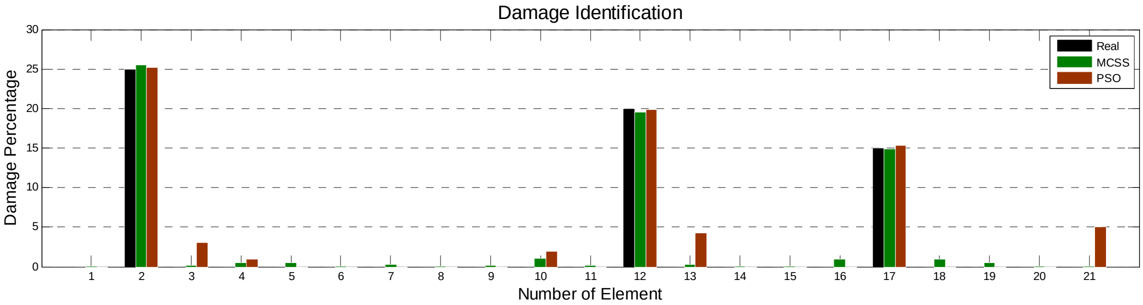


Fig. 27 Damage detection results of the algorithms for the frame (Scenario II)

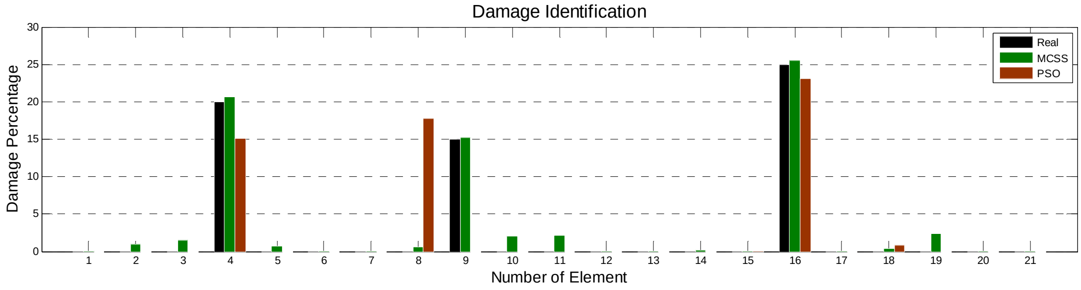


Fig. 28 Damage detection results of the algorithms for the frame (Scenario III)

6. Conclusions

The structural damage identification method based on natural frequencies and mode shapes are studied in this paper. In order to mitigate the effect of noise on input data, a penalty function is added to the objective function. MCSS and PSO algorithms are utilized to solve the optimization problem associated with the damage detection. In order to verify the performance of the proposed methodology, different numerical problems with different scenarios are tested. Although the proposed cost function in both algorithms performed adequately, in some of the scenarios PSO failed to find the correct scenario especially when the number of elements was increased, or the structure was more complicated. There are many local optimums in the objective function, which introduce difficulties in finding the global optimum; in this situation, it is necessary to use an optimization algorithm which has more exploration ability; since MCSS can escape from a local optimum in order to find the global optimum, in all of the examples it detects the correct scenario. From the results, it can be concluded that the proposed MCSS is quite efficient and robust for damage detection problems in a variety of structures.

Acknowledgements

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

References

Arumugam, M. and Rao, M. (2008), "On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems", *Appl. Soft Comput.*, **8**(1), 324-336.

Carden, E. and Fanning, P. (2004), "Vibration based condition monitoring: a review", *Struct. Health Monit.*, **3**(4), 355-377.

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

Esfandiari, A., Bakhtiari-Nejad, F., Rahai, A. and Sanayei, M. (2009), "Structural model updating using frequency response function and quasi-linear sensitivity equation", *J. Sound Vib.*, **326**(3), 557-573.

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

Friswell, M. and Mottershead, J. (2001), "Physical understanding of structures by model updating", *Proceedings of the COST F3 International Conference on Structural System Identification*.

Friswell, M., Penny, J. and Garvey, S. (1998), "A combined genetic and eigensensitivity algorithm for the location of damage in structures", *Comput. Struct.*, **69**(5), 547-556.

Kaveh, A. (2014), *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, Springer Verlag, Switzerland, May 2014.

Kaveh, A., Javadi, S.M. and Maniat, M. (2014), "Damage assessment via modal data with a mixed particle swarm strategy, ray optimizer, and harmony search", *Asian J. Civil Eng.*, **15**(1), 95-106.

Kaveh, A., Share, M. and Moslehi, M. (2013), "Magnetic charged system search: a new meta-heuristic algorithm for optimization", *Acta Mech.*, **224** (1), 85-107.

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

Kim, J.T., Ryu, Y.S., Cho, H.M. and Stubbs, N. (2003), "Damage identification in beam-type structures:

frequency-based method vs mode-shape-based method", *Eng. Struct.*, **25**(1), 57-67.

Liszkai, T. and Raich, A. (2005), "Solving inverse problems in structural damage identification using advanced genetic algorithm representations", *Proceeding of the 6th Congress of Structural and Multidisciplinary Optimization. ISSMO.*

Maity, D. and Tripathy, R. (2005), "Damage assessment of structures from changes in natural frequencies using genetic algorithm", *Struct. Eng. Mech.*, **19**, 21-42.

Majumdar, A., Maiti, D. 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.

Miguel, L., Miguel, L., Kaminski Jr, J. and Riera, J. (2012), "Damage detection under ambient vibration by harmony search algorithm", *Expert Syst. Appl.*, **39**(10), 9704-9714.

Parsopoulos, K. and Vrahatis, M. (2007), "Parameter selection and adaptation in unified particle swarm optimization", *Math. Comput. Model.,* **46**(1), 198-213.

Ratnaweera, A., Halgamuge, S. and Watson, H. (2004), "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients", *IEEE T. Evolut. Comput.*, **8**(3), 240-255.

Sahoo, B. and Maity, D. (2007), "Damage assessment of structures using hybrid neuro-genetic algorithm", *Appl. Soft Comput .*, **7**(1), 89-104.

Shi, Y. and Eberhart, R. (2001), "Fuzzy adaptive particle swarm optimization", *Proceedings of the 2001 Congress on Evolutionary Computation*.

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

Yang, Q.W. and Liu, J.K. (2007), "Structural damage identification based on residual force vector", *J. Sound Vib.*, **305**(1), 298-307.

CY