International Journal of
Mathematical Modelling & Computations
Vol. 13, No. 03, Summer 2023, 1- 11
DOI:10.30495/ijm2c.2023.1991903.1278

Damage Detection in Truss Structures Using Grasshopper Optimization Algorithm

P. Foroutanrad[a] and B. Adhami[a,*]

[a]*Department of Civil Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran.*

Abstract. Structural engineers' goal has constantly been identifying, restoring, repairing, or replacing damaged members. As a result, one of the most crucial and necessary steps in the upkeep and restoration of structures is identifying damaged members. Damage detection techniques from structural dynamic response measurements can often be used to detect and locate damage. This paper proposes a structural damage identification method based on changing natural frequency, finite element modeling, and the Grasshopper Optimization Algorithm (GOA). This algorithm mathematically models and mimics the behavior of grasshopper swarms in nature for solving optimization problems. As numerical examples, the 13-bar and a 31-bar planar truss are considered to examine the suggested methodology's precision. According to the findings, the recommended method is workable for systems with few members and minor damage. However, the accuracy of the diagnosed damage in structures with medium-sized members and considerable damages was poor, making it more likely to converge to local optimum points conditions.

Received: 25 July 2023, Revised: 07 September 2023, Accepted: 27 September 2023.

Keywords: Damage detection, Optimization algorithms, Truss structure, Objective function, Structural dynamic response, Natural frequency.

Index to information contained in this paper

1. Introduction

Despite the initial design methods, structures deteriorate over time and become damaged. This deterioration is due to various reasons, including damage caused by traffic loads, environmental factors (such as steel corrosion and concrete carbonation), and aging of construction materials. Additionally, structural damage can be caused by events such as earthquakes, hurricanes, and floods. Therefore, the health of a structure is influenced by operational and environmental factors, including normal load conditions, current and future environmental conditions, and expected hazards throughout its service life. These factors are uncertain variables, making it challenging to define structural health in terms of age, application, and safety level against severe natural reactions.

The damage leads to weakness in the overall behavior or one of the structure's members due to the applied loads, which affects the equations governing the system's movement. In this regard, by timely detecting and identifying damage in the early stages, an appropriate solution can be chosen at the right time. By repairing and renovating the structure, general deterioration and the resulting financial and life-threatening losses caused by structure

*Corresponding author. Email: beh.adhami@iauctb.ac.ir

https://sanad.iau.ir/journal/ijm

collapse can be prevented. Because if some of the structure's members are not identified and diagnosed correctly in time, they can cause catastrophic damage and fatal failures.

Based on the situation, the extent of damage, and the estimated safety level of the system, the decision will be made regarding the non-use or repair of the structure subjected to dynamic loads. The damages may spread over time, leading to overall damage and structural failure. Therefore, the issue of damage detection, as one of the fundamental topics in civil and mechanical engineering, has occupied the minds of many researchers [4,2].

On the one hand, methods for detecting damage can be categorized into four levels:

Level 1: Detection of damage, determining whether damage exists in the structure.

Level 2: Damage Localization, level 1 plus determining the geometric location of the damage.

Level 3: Damage Assessment, level 2 plus determining the amount of damage.

Level 4: Damage Prognosis, level 3 plus predicting the remaining service life of the structure.

In most studies, levels 1 to 3 have been investigated. Generally, the fourth level pertains to fracture mechanics, fatigue analysis, and reliability and is not addressed in structural dynamics [10].

Optimization is one of the critical tools in the decision-making and analysis of physical systems. Mathematically, an optimization problem is finding the best solution among a set of candidates or feasible solutions.

Optimization is a complicated subject in civil engineering, and various tools are available. Every process has the potential for improvement in itself, and this can be achieved by minimizing time, cost, and risk or maximizing profit, quality, and efficiency.

Based on previous research, finding an algorithm that works well in all optimization applications is impossible [8]. Therefore, various evolutionary algorithms with diverse search mechanisms have been proposed, and it has been shown that these algorithms can be effectively used in multiple problems. Among the evolutionary algorithms, different topics such as simplifying algorithms for use in all sciences or improving the search mechanism of evolutionary algorithms to increase the accuracy of obtaining optimal, or focusing on the convergence speed of evolutionary algorithms and avoiding the calculation of problems with computational costs in terms of processing and even runtime are considered. It is observed that different evolutionary algorithms have seen various improvements in these areas, and the combination of evolutionary algorithm operators is also investigated. The grasshopper optimization algorithm was proposed by Mirjalili et al. in 2017 and has shown promising results in obtaining global optimal in various functions in terms of complexity compared to algorithms presented in recent years, such as genetic algorithm, particle swarm optimization, bat algorithm, firefly algorithm, gravitational search algorithm, and flower pollination algorithm [12].

In recent decades, numerous methods have been proposed for identifying damage to structures. Initial plans were only capable of identifying the location of damage in structures. With the expansion of studies and the emergence of optimization-based methods, the ability to determine the severity of damage in structures has also become available. Most methods for identifying and diagnosing damage are based on changes in the structure's natural frequencies, changes in mode shapes, or the measurement of dynamic flexibility.

For the first time in 1966, metaheuristic algorithms were introduced with the proposal of evolutionary algorithms. Subsequently, studies were conducted to develop and improve these algorithms, leading to new algorithms based on living organisms in recent years. Based on population, well-known evolutionary algorithms include the ant colony optimization, the Artificial bee colony optimization, the particle swarm optimization, and the charged system search optimization [15]. The grasshopper optimization algorithm

(GOA) is one of the newest metaheuristic algorithms. This algorithm belongs to the swarm intelligence algorithms and is designed by taking inspiration from the social behavior of grasshoppers and how each grasshopper is influenced by its surrounding environment. In this algorithm, updating the position of each grasshopper is dependent on its distance from all grasshoppers in the current generation and the position of the best grasshopper. The characteristics of this algorithm include its simplicity and having only one adjustable parameter.

The following is a summary of several studies in the field of civil engineering:
Sun and Büyüköztürk investigated the optimal placement of sensors in frame and truss structures using the honey bee colony algorithm [14]. Mojtahedi and Baibordi examined damage detection using the particle swarm optimization algorithm and modal parameters of the structure [3]. Hosseini Vaezi et al. studied damage detection in steel shear walls using the wavelet algorithm [1]. Dizangian et al. utilized Mutation Teaching-Learning-Based Optimization (MTLBO) to predict the extent of damage in truss structures [6]. Additionally, Asnaashariya and Shayanfar investigated damage detection in structures using multi-objective optimization algorithms (NSGAII and MOPSO) and the VIKOR method [7]. Finally, Ding et al. employed the clustering-based tree seed algorithm to identify damage in structures with uncertain modeling errors and noise measurements [5]. Sahu and Nayak proposed a method for detecting damage in structural members using the adaptive genetic algorithm [11].

The present study focuses on identifying damage location in truss structures using the Grasshopper Optimization Algorithm (GOA) and the dynamic response of the structure (Natural Frequencies) induced by free vibration.

2. Proposed methodology

One of the consequences of structural damage is a reduction in the stiffness of the member, which is well demonstrated by changes in the structure's natural frequency. Moreover, the occurrence of damage leads to an increase in structural damping. Implementing these factors in the optimization method makes it possible to detect the damaged element accurately. In this study, using the proposed method, analysis was performed using the finite element method and programming in MATLAB software. Accordingly, a mathematical model of the structure was programmed in MATLAB software, and modal analysis was used to obtain the structure's natural frequency. To achieve this objective, the relevant issue has been addressed by coding and obtaining the stiffness and mass matrices of the structure under discussion. Subsequently, based on the principles of structural dynamics, the natural frequencies of the intended structures have been calculated. Then, utilizing an objective function and the natural frequencies of both intact and damaged structures, the problem has been formulated as an optimization problem.

2.1 *The objective function*

It is defined as a function or expression that represents the objective of the problem. In optimization, selecting an appropriate objective function is one of the most crucial decisions. In optimization problems, the aim is to minimize the value of this function. The objective function should reflect the primary objective of the optimization problem. In this study, the Efficient Correlation-Based Index (ECBI) introduced by Seyedpoor and Nobahari in 2011 has been chosen as the objective function. This function combines two proposed functions, MDLAC and obj. The MDLAC function, expressed in equation (1), compares two frequency change vectors, one obtained from the tested structure and the other from the desired structure's analytical model (modal analysis). The MDLAC index is susceptible to identifying damaged elements as the objective function. However, it may have less sensitivity in identifying healthy elements, such that this function correctly identifies the location of damaged elements but may also identify healthy elements as damaged [13].

$$MDLAC(X) = -\frac{|\Delta F^T.\delta F(X)|^2}{(\Delta F^T.\Delta F)\big(\delta F^T(X).\delta F(X)\big)} \tag{1}$$

The obj function, defined in equation (2), is a frequency-based index for the structure. Compared to the MDLAC index, this function identifies healthy elements more quickly. Still, it is less sensitive in identifying damaged elements and has a higher probability of misclassifying a damaged element as healthy [13].

$$obj(X) = \frac{1}{n_f}\sum_{i=1}^{n_f}\frac{min(f_{xi}‘f_{di})}{max(f_{xi}‘f_{di})} \tag{2}$$

As mentioned, by combining these two functions, a new function called ECBI is defined according to equation (3).

$$ECBI = -\frac{1}{2}[MDLAC(X) + obj(X)] \tag{3}$$

2.2 *Introducing the GOA*

The Grasshopper Optimization Algorithm (GOA) is one of the latest metaheuristic algorithms. This algorithm belongs to the category of swarm intelligence algorithms. It was designed by Seyedali Mirjalili et al. in 2017, inspired by the social behavior of grasshoppers and how each grasshopper is influenced by its surroundings. In this algorithm, the update of each grasshopper's position depends on the distance of that grasshopper to all other grasshoppers in the current generation and the position of the best grasshopper. One of the notable features of this algorithm is its simplicity and having only one adjustable parameter. The mathematical model used to simulate the behavior of grasshoppers is initially represented by equation (4) [12].

$$X_i = S_i + G_i + A_i \tag{4}$$

In this equation, X_i represents the position of the *i*-th grasshoppers, S_i denotes the social interaction, and G_i is the gravitational force acting on grasshopper *i*-th. A_i represents the wind direction. To create random behavior, the above equation can be rewritten as equation (5), in which r_1, r_2, and r_3 are random numbers in the interval [0,1].

$$X_i = r_1S_i + r_2G_i + r_3A_i \tag{5}$$

The value of social interaction for grasshopper *i*-th, denoted by *S*, is calculated based on equation (6).

$$S_i = \sum_{\substack{j=1\\ j\neq i}}^{N} s(\,d_{ij})\widehat{d_{\iota J}} \tag{6}$$

Which d_{ij} indicates the distance between the *i*-th and *j*-th grasshoppers and is calculated by equation (7):

$$d_{ij} = |x_{j-}x_i| \tag{7}$$

A function defines social interaction pressure, as shown in equation (8), where d_{ij} is a unit vector from the *i*-th grasshopper to the *j*-th grasshopper, given by $\widehat{d_{\iota J}} = \frac{x_{j-}x_i}{d_{ij}}$.

The function S, which defines the social force, is calculated as per equation (8), where *f* represents the intensity of gravity and *l* represents the gravity scale length.

$$S(r) = fe^{\frac{-r}{l}} - e^{-r} \tag{8}$$

The components *G* and *A* in equation (4) are also expressed as per equations (9) and (10), where *g* is the gravitational constant, and $\widehat{e_g}$ is a unit vector towards the center of the earth, *u* is the sliding constant, and $\widehat{e_w}$ is a unit vector in the direction of the wind. Nymph grasshoppers have no wings; hence their movement is highly dependent on the direction of the wind.

$$G_i = -g\widehat{e_g} \tag{9}$$

$$A_i = u\widehat{e_w} \tag{10}$$

By substituting *S*, *G*, and *A*, equation (4) can be expanded into equation (11), where $S(r) = fe^{\frac{-r}{l}} - e^{-r}$ *and N is the* number of grasshoppers.

$$X_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} s(|X_j - X_i|)\frac{X_j - X_i}{d_{ij}} - g\widehat{e_g} + u\widehat{e_w} \tag{11}$$

However, this mathematical model cannot be directly used to solve optimization problems, mainly because the grasshoppers quickly settle in an area where the attraction and repulsion forces of other grasshoppers in that area are equal, which is called the comfort zone in this field. As a result, the group does not converge to a specific point. This area is shown in Figure 1. The modified version of this equation for solving optimization problems is given by equation (12).

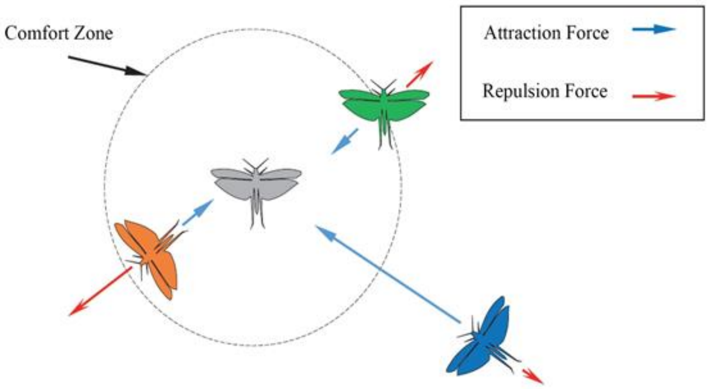

Figure 1. Illustrates the grasshopper's comfort zone, attraction, and repulsion forces in this area.

In this equation, ub_d is the upper bound at dimension *d*-th, lb_d is the lower bound at dimension *d*-th, and T_d is the value at dimension *d*-th in the target (best solution seen so far). *c* is a constant that reduces the comfort zone, repulsion, and attraction. In this equation, *S* is approximately the same as the *S* component in equation (4); however, the gravitational parameter (*G*) and wind direction (*A*) are not considered. We assume that the wind direction is always toward the target.

$$X_i^d = c\left(\sum_{\substack{j=1\\ j\neq i}}^{N} c\frac{ub_d - lb_d}{2} s\left(\left|X_j^d - X_i^d\right|\right)\frac{X_j - X_i}{d_{ij}}\right) + \widehat{T_d} \tag{12}$$

Equation (12) shows that the next position of a grasshopper is defined based on its current position, the target position, and the positions of all other grasshoppers. Note that the first component in this equation is the position of the current grasshopper concerning the other grasshoppers. We consider the positions of all grasshoppers to define the search agents' positions around the target.

The component must decrease with increasing iterations during the algorithm to maintain a balance between exploration and exploitation. This method strengthens the exploration by increasing the number of iterations. The coefficient that reduces the comfort zone proportional to the number of iterations is calculated using equation (13).

$$c = cmax - l\frac{cmax - cmin}{L} \tag{13}$$

In this equation, the mathematical model of the algorithm is considered, where c_{max} represents the maximum value, c_{min} represents the minimum value, and l indicates the current number of iterations. Additionally, L represents the maximum number of iterations of the algorithm.
In this study, c_{max} equals1 and, c_{min} is set to 0.00001.

The concepts above suggest that the mathematical model of the algorithm needs the grasshoppers to gradually converge towards the objective during the iterations of the algorithm. However, there is no objective in real search space as we need to know the exact global optimum location. Therefore, we will find a goal for the grasshoppers in each optimization stage.

The algorithm can be expressed as follows:

1. First, each member of the population takes a position in the space of feasible solutions.
2. The variables c_{max} , c_{min}, and the maximum number of iterations are initialized.
3. The objective function value is calculated for all population members.
4. The grasshoppers are sorted based on the objective function's value, the population's best member is identified, and then the algorithm's main loop starts.
5. The value of c is determined (it will not change in subsequent iterations and will update for the next replications).
6. For each grasshopper, a new position is determined based on the equations.
7. The grasshoppers are evaluated.
8. The best member of the population is updated.
9. If the termination conditions are not met, we return to step 5; otherwise, the algorithm terminates.

3. Damage detection method using GOA

In this study, the Grasshopper Optimization Algorithm (GOA) was employed to address the damage identification problem in truss structures. The steps for damage identification using the algorithm mentioned above are described as follows:
i) Structural analysis and calculation of natural frequencies of the structure of interest.
ii) Determination of natural frequencies of healthy and damaged structures.
iii) Definition and formation of the objective function.
iv) Implement the GOA algorithm to minimize the objective function formed in the

previous step.

4. Numerical examples

The presented findings aim to assess the algorithm's ability and methodology for identifying damage in this section. To this end, the results of two numerical examples comprising 13 and 31-member trusses, each with two damage cases, are provided for examination.

4.1 *Thirteen element planar truss*

The finite element model of a thirteen-element truss structure is shown in Figure 2. This model consists of eight nodes and thirteen members. In this example, the damage is modeled by reducing the elasticity modulus, and the first ten natural frequencies of the structure are used to detect the damage. In all elements, the density equals ρ=7850 kg/m^3, the elasticity modulus equals E=200 GPa, and the length equals L=2 m.
The cross-sectional area of all elements is taken as 0.01 m^2. It should be noted that any rational number can be assumed for the cross-sectional area, as it does not affect the nature of the task, identifying the damage.

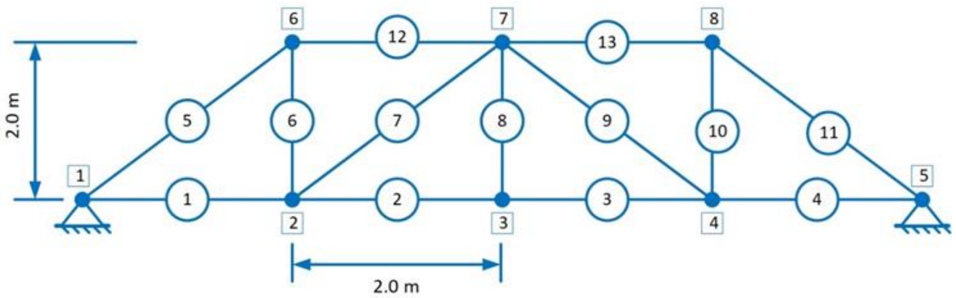

Figure 2. The 13-bar planar truss, along with its element and node numbers.

According to Table 1, two cases of damage have been considered for this structure, and the results of identifying the damage are also presented in Figures 3 and 4.

Table 1. Considered damage cases for 13-bar planar truss.

Damage case 1		Damage case 2	
Element number	Damage ratio	Element number	Damage ratio
7	0.2	5	0.1
-	-	8	0.25
-	-	12	0.15

4.2 *Thirty-one element planar truss*

The finite element model of a thirteen-element truss structure is shown in Figure 5. The mentioned model is composed of 14 nodes with 25 degrees of freedom and 31 elements. In this example, the damage is modeled by reducing the elasticity modulus and using the first ten natural frequencies of the structure to detect the damage. In all elements, the density equals ρ=2770 kg/m3, the elasticity modulus equals E=70 GPa, and the length equals L= 1.52 m [9].

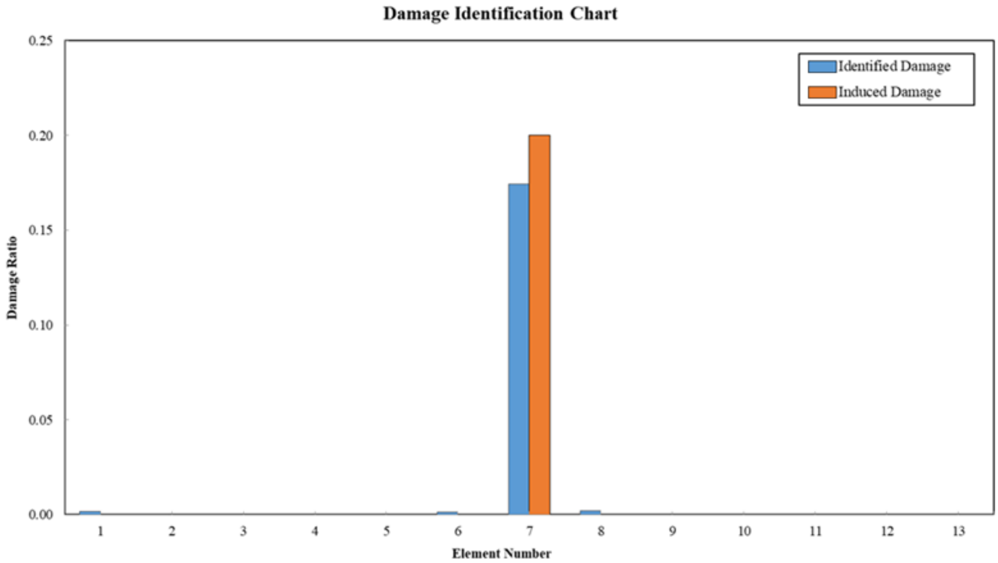

Figure 3. Detection of damage in 13-element planar truss (Case 1).

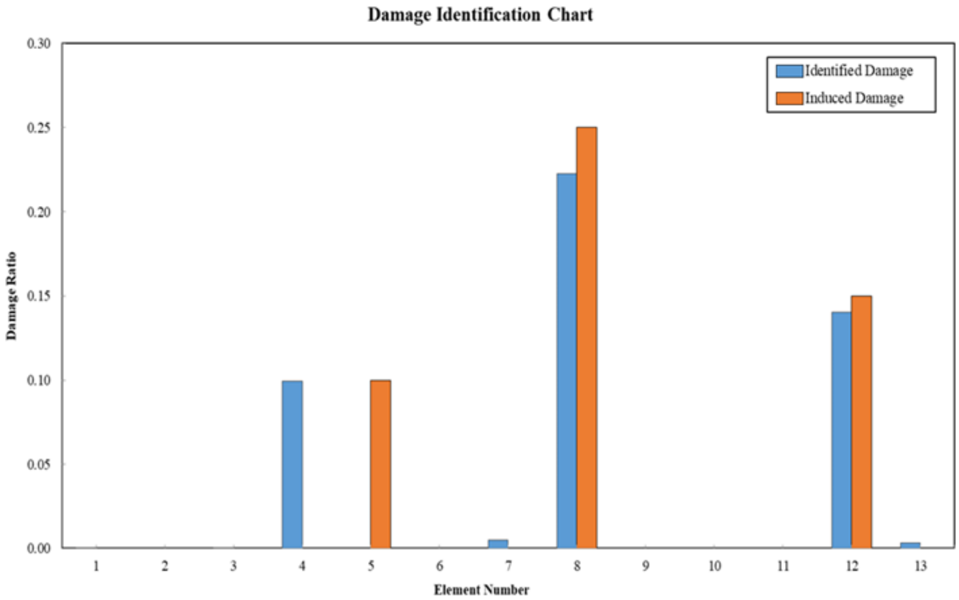

Figure 4. Detection of damage in 13-element planar truss (Case 2).

The cross-sectional area of all elements is assumed to be 25 cm^2. It should be noted that any rational number can be assumed for the cross-sectional area, as it does not affect the nature of the task, identifying the damage.

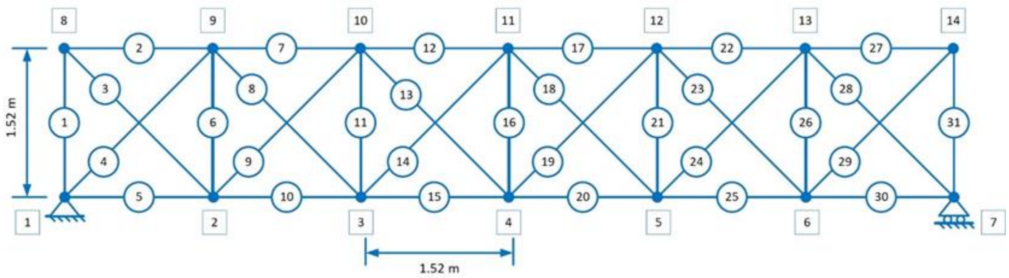

Figure 5. The 31-bar planar truss, along with its element and node numbers.

According to Table 2, two damage cases have been considered for this structure, and the results related to identifying its damage have also been presented in Figures 6 and 7.

Table 2. Considered damage cases for 31-bar planar truss.

Damage case 1		Damage case 2	
Element number	Damage ratio	Element number	Damage ratio
16	0.3	11	0.25
-	-	25	0.15

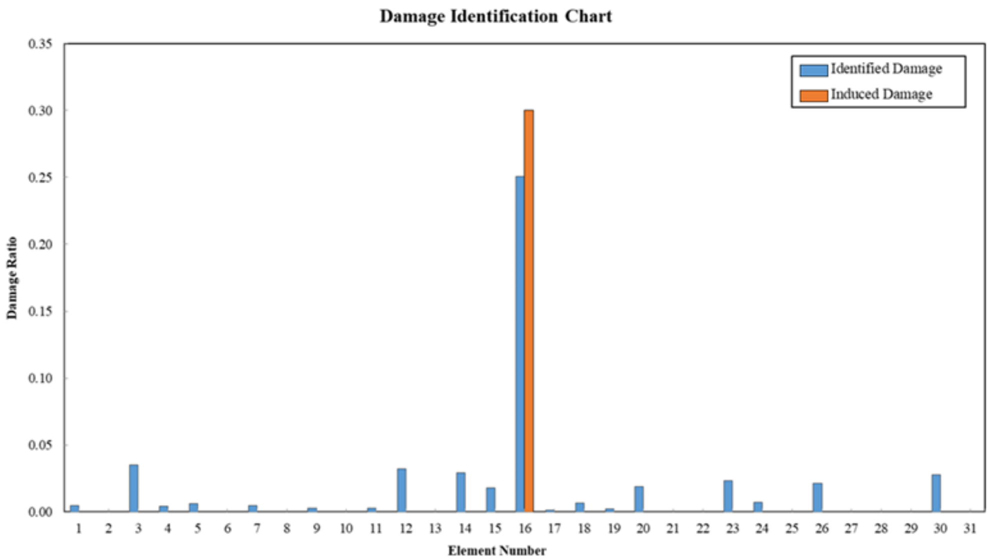

Figure 6. Detection of damage in 31-element planar truss (Case 1).

As seen in Figures 3, 4, 6, and 7, the algorithm identified the location and extent of damage in a 13-bar planar truss with acceptable accuracy. However, in the 31-bar planar truss, as the number of truss elements increased and the number of damaged elements, although the location and extent of damage were identified, the accuracy of the results decreased. Damage was incorrectly detected in several other locations. This is due to the influence of damage on different elements and the algorithm getting stuck in local optimum points.

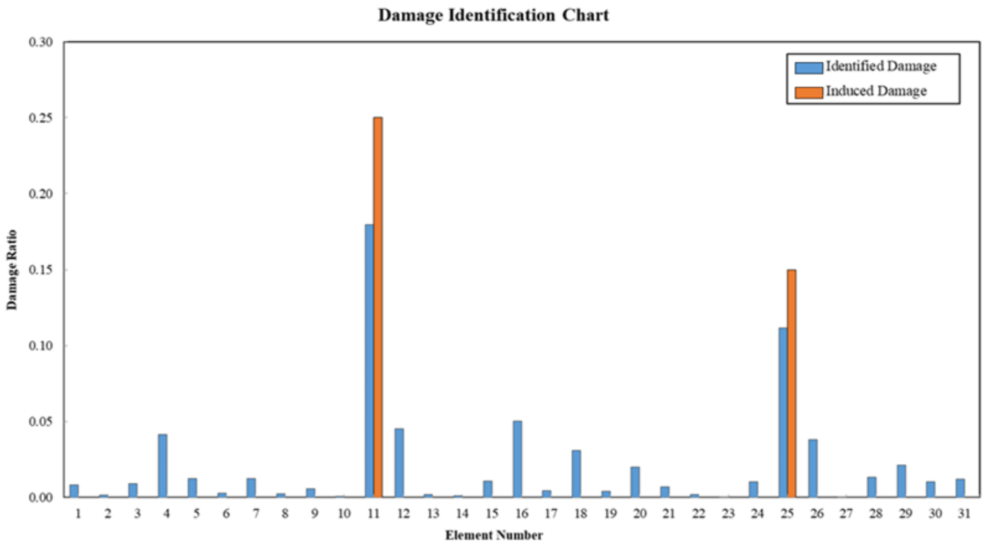

Figure 7. Detection of damage in 31-element planar truss (Case 2).

5. Conclusion

The type of algorithm, objective function, presence or absence of noise, the number of structural elements, and the number of damages significantly impact the accuracy of the results and the speed of providing answers. Based on the numerical examples investigated, it was observed that the results obtained for the 13-element planar truss structure had an acceptable accuracy. However, in the 31-element planar truss structure, due to the increase in the number of damaged elements and structural elements and the vast search space, the accuracy and ability of the algorithm decreased, and the desired and satisfactory results were not achieved. Therefore, the identified damages in the medium-sized truss structure and multiple damages were affected by low accuracy, causing the algorithm to fall into local optimum points. In this regard, the ability of the algorithm, as mentioned earlier, should be investigated using methods with higher dynamic characteristics.

References

[1] A. Abbasi, S. R. Hosseini Vaez and G. Ghodrati Amiri, Determining the damage of steel shear wall using wavelet algorithm, Ferdowsi Civil Engineering Journal, **2 (29)** (2017) 61–74.

[2] B. Adhami, et al., Detection of damage to bending structures using a crack index, Structures & Buildings, **167** (2012) 178–186.

[3] S. Baybordi and A. Mojtahedi, Development of an efficient method of structural damage diagnosis in templated platforms using the meta-heuristic algorithm of particle assembly, Journal of Civil and Environmental Engineering of Tabriz University, **46.4 (85)** (2017) 37–49.

[4] H.-P. Chen and Y.-Q. Ni, Structural Health Monitoring of Large Civil Engineering Structures, John Wiley & Sons, (2018).

[5] Z. Ding, et al., Structural damage identification with uncertain modeling error and measurement noise by clustering-based tree seeds algorithm, Engineering Structures, **185** (2019) 301–314.

[6] B. Dizangian, M. Ghasemi and M. Ghasemi, Damage diagnosis of truss structures based on vibration data using the learning-based optimization method (TLBO), 6th National Congress on Civil Engineering, Architecture, and Urban Development, (2019).

[7] E. Ethnasharieh and M. A. Shayanfar, Identification of damage in structures with the help of multi-objective optimization algorithms NSGAII and MOPSO using Vicor method, 3rd International Conference on Applied Research in Structural Engineering and Construction Management, (2019).

[8] D. Lim, et al., Generalizing surrogate-assisted evolutionary computation, IEEE Transactions on Evolutionary Computation, **14 (3)** (2010) 329–355.

[9] A. Messina, E. J. Williams and T. Contursi, Structural damage detection by a sensitivity and statistical-based method. journal of sound and vibration, **216 (5)** (1998) 791–808.

[10] A. Rytter, et al., Vibrational based inspection of civil engineering structures, Aalborg University, (1993).

[11] S. Sahu, B. B. Nayak, (2020), An adaptive genetic algorithm method for damage detection in structural elements, Materials Today: Proceedings, **26** (2020) 581–585.

[12] S. Saremi, S. Mirjalili and A. Lewis, Grasshopper optimization algorithm: theory and application, Advances in Engineering Software, **105** (2017) 30–47.

[13] S. M. Seyedpoor and M. Nobahari, Structural damage detection using an efficient correlation-based index and a modified genetic algorithm, Mathematical and Computer Modelling, **53 (9-10)** (2011) 1798–1809.

[14] H. Sun and O. Büyüköztürk, Optimal sensor placement in structural health monitoring using discrete optimization, Smart Materials and Structures, **24 (12)** (2015) 125034.

[15] M. Talebpour and V. Kalat-Jari, Metaheuristic combined algorithm for the optimization design of steel structures, Omran Sharif Engineering Journal, (2017).