Damage identification in steel plate using FRF and inverse analysis

Samir Khatir, Magd Abdel Wahab
Soete Laboratory, Faculty of Engineering and Architecture, Ghent University, Technologiepark Zwijnaarde 903, B-9052, Zwijnaarde, Belgium
Khatir_samir@hotmail.fr, https://orcid.org/0000-0002-8101-3633
Magd.AbdelWahab@UGent.be, https://orcid.org/0000-0002-3610-865X

Samir Tiachacht
Laboratory of Mechanics, Structure and Energetics (LMSE), Mouloud Mammeri University of Tizi-Ouzou, B.P.N°17 RP, 15000, Algeria

Cuong Le Thanh
Ho Chi Minh City Open University, Ho Chi Minh City, Viet Nam

Roberto Capozucca, Erica Magagnini
Università Politecnica delle Marche, Ancona, Italy

Brahim Benaissa
Toyota Technological Institute, Department of Mechanical Systems Engineering, Design Engineering Lab, 468-8511 Aichi, Nagoya, Tempaku Ward, Hisakata, 2 Chome−12−1, Japan
https://orcid.org/0000-0002-9472-9331

ABSTRACT. Metaheuristic algorithms have known vast development in recent years. And their applicability in engineering projects is constantly growing; however, their deferent exploration and exploitation techniques cause the engineering problems to favor some algorithms over others. This paper studies damage identification in steel plates using Frequency Response Function (FRF) damage indicator to detect and localize the healthy and damaged structure. The study is formulated as an inverse analysis, investigating the performance of three new metaheuristic algorithms of Wild Horse Optimizer (WHO), Harris Hawks Optimization (HHO), and Arithmetic Optimization Algorithm (AOA). The objective function is based on measured and calculated FRF damage indicators. The results showed that the case of four damages with different damage severity levels presented a good challenge where the HWO algorithm was shown to have the best performance. Both in convergence speed and CPU time.

INTRODUCTION

Metaheuristic optimization algorithms are very robust tools that help solve almost any problem with the input-output relationship. They require a limited number of control parameters, depending on the algorithm strategy [1]. Some require several parameters, while other algorithms need just the population size and the search parameters. These are the search boundaries of design variables and the search stopping criteria; most studies use a maximum number of iterations [2].

The population size is the number of potential solutions considered in every iteration; following the structure of early evolutionary algorithms, the strategy of the population is adopted in most metaheuristic algorithms because of its efficiency. Each potential solution is a set of values for the problem parameter; the algorithm suggests them within the earlier set boundaries. Depending on the performance of each set, the new sets for the population are calculated according to the algorithm's design [3].

The performance of the potential solutions is compared based on their fitness value. In a minimization problem, the set of variables corresponding to the minimum fitness are considered to be the best solution. The algorithm tries to reach a better solution in each iteration by generating new solutions based on the earlier knowledge. Most metaheuristics have two strategies; the first strategy is exploration, when the algorithm is looking to find solutions in different areas of the design space. This feature makes metaheuristics algorithms a powerful tool against non-convex problems and problems with many local minima. The other strategy is called exploitation, which is when the algorithm locks on a small area and looks deeper to find better solutions that are very close. This strategy allows these algorithms to find precise solutions and compete for accuracy with classical methods like the gradient descent method [2, 3].

The application of metaheuristic algorithms in structural health monitoring is widely adopted due to their good performance in such methods. Due to the complexity of structural behavior, most research projects are formulated as ill-posed inverse problems, both in the behavior nonlinearity of the material and in the behavioral characteristics of the structures. Classical methods cannot solve such ill-posed problems, and they often require complex analytical methodologies. Tiachacht et al. [4] investigated Genetic Algorithm (GA) for crack identification in 3D. Zenzen et al. [5, 6] suggested Bat optimization algorithm in damage detection in truss structures. Cuong-le et al. [7] suggested a PSO and Support Vector Machine (SVM) method for structural health monitoring. Khatir et al. used Jaya algorithm to predict crack size and orientation in steel plates [8]. Chen et al. [9] presented a hybrid Nelder–Mead algorithm (NM) Ant lion optimizer (ALO) for the estimation of multiple damages. Livani et al. [10] proposed an enhanced particle swarm optimization (PSO) with a strategy called active/inactive flaw (AIF) of damage identification in an Aluminum plate. In [11] The Cuckoo search (CS) algorithm merged with PSO was suggested to predict structural damages under temperature variation.

Several metaheuristic algorithms exist, and although they all offer better performance in ill-posed problems, each engineering problem is unique in how output varies according to the change in the design variables. Therefore, some metaheuristic methods perform better than others in different areas. This is because each optimization method has a different way to calculate the variables of every iteration and different ways to change from exploration to exploitation. Which can make some inverse problems favor some strategies over others. Benaissa et al. [12-14] compared several optimization algorithms for the suggested method of fast crack identification in steel plates. Mishra et al. [15] compared the performance of 10 optimization algorithms in damage detection in trusses. Ding et al. compared the Jaya and Tree Seeds Algorithm [16] in the case of experimental and simulation-based damage estimation. Moezi et al. [17] proposed an improved CS algorithm. Its performance is compared to well-established optimization algorithms.

The AOA [18] is a method based on mathematical primes, using a Math Optimizer Accelerated strategy, a function that helps explore the search space. It uses another technique called the Math Optimizer probability, which is a function that helps guide the search. The HHO [19] uses a strategy inspired by the Harris' hawks. Namely, the hunting behavior, as wait to detect a prey based on two systems. The soft besiege and the hard besiege, in combination with the rapid dive. These techniques are modeled mathematically, and each potential solution is considered a hawk in this algorithm, looking to find better solutions in every iteration by reimplementing the search techniques. The WHO optimization algorithm is a metaheuristic presented by Iraj et al. [20] inspired by the social life of wild horses. It models the hierarchical characteristics for selecting and changing the leader. This method is led by what is called grazing behavior. It guides the horse herd members to search in different radiuses around the leader.

FRF is a damage indicator based on the vibrational response of the structure and calculated from its mass and stiffness matrices [21]. This paper compares the performance of WHO, HHO, and AOA algorithms in damage identification in a CCCC rectangular plate. Based on the FRF indicator. The plate is discretized into 100 elements, and the goal for the algorithm is to predict what element is damaged and to what degree it is damaged. The algorithms are subjected to the same search conditions, run on the same computer, and using the same search parameters. The second section discusses the RFR damage indicator theory and how it is calculated; this indicator will be used later by the optimization algorithms for the fitness evaluation. The third section presents the details of each optimization algorithm. Section 4 shows the considered damage scenarios and discusses the results obtained using each algorithm. We compare their performance in the case of limited damage element and damage severity variables and a final case with four damaged elements with different damage severity. And evaluating their computational cost. The results show that there is one metaheuristic algorithm that performs better than the others in all cases.

FRF FOR DAMAGE INDEX

The following formulation present FRF for the healthy and damaged structure:

$$
\begin{cases}
\left[\mathrm{H}(\omega) \right]^{A} = \left(-\omega^2 \left[M \right]^{A} + \left[K \right]^{A} \right)^{-1} \\
\left[\mathrm{H}(\omega) \right]^{T} = \left(-\omega^2 \left[M \right]^{T} + \left[K \right]^{T} \right)^{-1}
\end{cases}
\tag{1}
$$

where $[M]$ and $[K]$ are mass and stiffness matrices, the symbol $[\ ]^{A}, [\ ]^{T}$ are undamaged and damaged cases, respectively. As a result of the damage, the stiffness changes as follows:

$$
\left[\Delta K \right] = \left[K \right]^{A} - \left[K \right]^{T}
\tag{2}
$$

where $[K]^{A}, [K]^{T}$ denotes the stiffness healthy and damaged structure, respectively
By combining Eqns. (1) and (2), we can write:

$$
\left[\Delta K \right] = \left(\left[\mathrm{H}(\omega) \right]^{A} \right)^{-1} - \left(\left[\mathrm{H}(\omega) \right]^{T} \right)^{-1}
\tag{3}
$$

η denote values for the position of damage considering the degree of freedom and position of elements and by using the first row of the global FRF matrix are defined as:

$$
\eta(1,i) = \left[\mathrm{H}(\omega) \right]^{T}_{1n} \left(\left[\mathrm{H}(\omega) \right]^{A} \right)^{-1}_{(:,i)} - \left[I \right]_{(:,i)}
\tag{4}
$$

In order to be able to develop the new formulation-based damage index, you should first learn how to write a formula, then the dimensions of the vector to the dimension of total degrees of freedom (*dofs*) should be increased, including the boundary conditions. The damage index can be written as follows [22]:

$$
\beta_i = \sqrt{\sum_{i}^{N} \left| \eta \left(dof_i \right)_j \right|}
\tag{5}
$$

OPTIMIZATION

For the purposes of quantification of localized damages by the Frequency Response Function (FRF) damage index method, we present in this section three optimization methods that we are going to use, namely: Harris hawks

optimization (HHO) [19], Arithmetic optimization algorithm (AOA) [18] and Horse herd optimization algorithm [20].

Harris hawks optimization (HHO) [19]
- *Exploration phase*

In HHO [19], the Harris' hawks perch randomly on some locations and wait to detect the prey based on two strategies.

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 |X_{rand}(t) - 2r_2 X(t)| & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \tag{6}$$

Where $X(t+1)$ is the position vector of hawks in the next iteration t, $X_{rabbit}(t)$ is the position of rabbit, $X(t)$ is the current position vector of hawks, r_1, r_2, r_3, r_4 and q are random numbers inside (0,1), which are updated in each iteration, LB and UB show the upper and lower bounds of variables, $X_{rand}(t)$ is a randomly selected hawk from the current population, and X_m is the average position of the current population of hawks. The average position of hawks is attained using Eqn. (7):

$$X_m(t) = \frac{1}{N_b} \sum_{i=1}^{N_b} X_i(t) \tag{7}$$

$X_i(t)$ Indicates the location of each hawk in iteration t and N_b denotes the total number of hawks.

- *Transition from exploration to exploitation*

To model this step, the energy of a rabbit is modeled as:

$$W = 2W_0\left(1 - \frac{t}{T}\right) \tag{8}$$

W indicates the escaping energy of the prey, T is the maximum number of iterations and W_0 is the initial state of its energy.

- *Exploitation phase*

This behavior is modeled by the following rules:

$$X(t+1) = \Delta X(t) - W|JX_{rabbit}(t) - X(t)| \tag{9}$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \tag{10}$$

where $\Delta X(t)$ is the difference between the position vector of the rabbit and the current location in iteration t, r_5 is a random number inside (0,1), and $J = 2(1 - r_5)$ represents the random jump strength of the rabbit throughout the escaping procedure. The J value changes randomly in each iteration to simulate the nature of rabbit motions.
In this situation, the current positions are updated using Eqn. (11):

$$X(t+1) = X_{rabbit}(t) - W|\Delta X(t)| \tag{11}$$

To perform a soft besiege, we supposed that the hawks can evaluate (decide) their next move based on the following rule in Eqn. (12):

$$Y = X_{rabbit}(t) - W|JX_{rabbit}(t) - X(t)| \tag{12}$$

We supposed that they dive based on the LF-based patterns using the following rule:

$$Z = Y + S \times LF(D) \tag{13}$$

where D is the dimension of the problem and S is a random vector by size $1 \times D$ and LF is the levy flight function, which is calculated using Eqn. (14):

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \tag{14}$$

where u, v are random values inside $(0,1)$, β is a default constant set to 1.5.

Hence, the final strategy for updating the positions of hawks in the soft besiege phase can be performed by Eqn. (15):

$$X(t+1) = \begin{cases} Y & \text{if } F(Y) > F(X(t)) \\ Z & \text{if } F(Z) > F(X(t)) \end{cases} \tag{15}$$

where Y and Z are obtained using Eqns. (12) and (13).

The following rule is performed in hard besiege condition by Eqn. (15); where Y and Z are obtained using new rules in Eqns. (16) and (17).

$$Y = X_{rabbit}(t) - W \left| J X_{rabbit}(t) - X_m(t) \right| \tag{16}$$

$$Z = Y + S \times LF(D) \tag{17}$$

where $X_m(t)$ is obtained using Eqn. (7).

Arithmetic optimization algorithm (AOA) [18]

As presented in the following formulation Eqn. 18, the optimization process begins with a collection of candidate solutions (X) which would be created at random, and in each iteration, the best candidate solution is assumed the best-obtained solution or approximately the optimum so far.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & x_{2,j} & \cdots & x_{2,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N-1,1} & \cdots & x_{N-1,j} & \vdots & x_{N-1,n} \\ x_{N,1} & \cdots & x_{N,j} & x_{N,n-1} & x_{N,n} \end{bmatrix} \tag{18}$$

Eqn. (19) compute the Math Optimizer Accelerated (MOA) feature used in the main search phrases.

$$MOA(C_{Iter}) = Min + C_{Iter} \times \left(\frac{Max - Min}{M_{Iter}} \right) \tag{19}$$

where $MOA(C_{Iter})$ presents the function value at the tth iteration, which is figured out by Eqn. (19). C_{Iter} indicates the current iteration, which is between $[1 \sim M_{Iter}]$. *Min* and *Max* the accelerated function's minimum and maximum values, respectively.

The most straightforward rule, which can simulate the action of Arithmetic operators, has been presented. The following position updating calculations for the exploration sections are shown in the following formulation:

$$x_{i,j}(C_{Iter}+1) = \begin{cases} best(x_j) \div (MOP + \varepsilon) \times ((UB_j - LB_j) \times \mu + LB_j) & r_2 < 0.5 \\ best(x_j) \times MOP \times ((UB_j - LB_j) \times \mu + LB_j) & \text{Otherwise} \end{cases} \quad (20)$$

where $x_i(C_{Iter}+1)$ denotes the ith solution in the next iteration, $x_{i,j}(C_{Iter})$ represents the jth position of the ith solution at the current iteration, and $best(x_j)$ is the jth position in the best-obtained solution so far. ε is a small integer number, UB_j and LB_j represents the upper and lower values of the jth position, respectively. μ indicates a control parameter to adapt the search process, can be presented by 0.5 based on the analyzed problem.

$$MOP(C_{Iter}) = 1 - \frac{C_{Iter}^{1/\alpha}}{M_{Iter}^{1/\alpha}} \quad (21)$$

Math Optimizer probability is used as a coefficient, represents the function value at the tth iteration, (C_{Iter}) indicates the current iteration, and (M_{Iter}) indicates the maximum number of iterations. α denotes a sensitive parameter and defines the exploitation accuracy over the iterations, which is fixed equal to 5.

The MOA function value conditions this phase of searching for the condition of r1 is not greater than the current $MOP(C_{Iter})$ value (see Eqn. (18)). In AOA, the exploitation operators (Subtraction (S) and Addition (A)) of AOA Explore the search field in-depth on many dense regions and use two fundamental search techniques to come up with a better solution that is modeled.

$$x_{i,j}(C_{Iter}+1) = \begin{cases} best(x_j) - MOP \times ((UB_j - LB_j) \times \mu + LB_j) & r_3 < 0.5 \\ best(x_j) + MOP \times ((UB_j - LB_j) \times \mu + LB_j) & \text{Otherwise} \end{cases} \quad (22)$$

The AOA is a relatively new algorithm, and this work is one of the early attempts to test its performance in damage quantification.

Horse herd optimization algorithm [20]
Iraj and Farshid [20] present a new optimizer algorithm called the wild horse optimizer (WHO), inspired by the social life behavior of wild horses.
The wild horse optimizer consists of five main steps as follows:

- *Creating an initial population and forming horse groups, and selecting leaders;*

The basic framework of all optimization algorithms is the same. The algorithm starts with $(\vec{x}) = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n\}$ an initial random population. The target function repeatedly evaluates this random population, and a target value is determined $(\vec{O}) = \{O_1, O_2, \dots, O_n\}$. It is also improved by a set of rules that are the core of an optimization technique.

- *Grazing and mating of horses;*

Eqn. (23) simulates grazing behavior. Eqn. (23) causes group members to move and search around the leader with a different radius.

$$\overline{X}_{i,G}^{j} = 2Z\cos\left(2\pi RZ\right)\times\left(Stallion^{j} - X_{i,G}^{j}\right) + Stallion^{j} \qquad (23)$$

$X_{i,G}^{j}$ is the current position of the group member (foal or mare), $Stallion^{j}$ is the position of the stallion (group leader), Z is an adaptive mechanism calculated by Eqn. (24), R is a uniform random number in the range $[-2,2]$ that causes The Grazing of horses at different angles (360 degrees) of the group leader, and finally $\overline{X}_{i,G}^{j}$ is the new position of the group member when grazing.

$$
\begin{aligned}
P &= \vec{R}_1 < TDR \\
IDX &= \left(P == 0\right) \\
Z &= R_2 \Theta IDX + \vec{R}_3 \Theta\left(\sim IDX\right)
\end{aligned}
\qquad (24)
$$

P is a vector consisting of 0 and 1 equal to the dimensions of the problem, $\vec{R}_1$ and $\vec{R}_3$ are random vectors with uniform distribution in the range $[0,1]$, R_2 is a random number with uniform distribution in the range $[0,1]$, IDX indexes of the random vector $\vec{R}_1$ returns that satisfy the condition $\left(P == 0\right)$.

APPLICATION

I n this section, we consider a fully clamped (CCCC) square plate (side a) with a thickness-to-side ratio $h/a = 0.01$. The non-dimensional natural frequency is given by

$$\overline{\omega} = \omega_{mn} a \sqrt{\frac{\rho}{G}} \qquad (25)$$

Frequency	[23]	[24]	[25] 20 × 20 Q4	Actual 10 × 10 Q4	Damage case		
					1	2	3
1	1.594	1.5582	1.5955	1.6216	1.6140	1.6101	1.5849
2	3.039	3.0182	3.0662	3.1893	3.1603	3.1329	3.0609
3	3.039	3.0182	3.0662	3.1893	3.1814	3.1778	3.0888
4	4.265	4.1711	4.2924	4.4458	4.4287	4.3580	4.2329
5	5.035	5.1218	5.1232	5.5280	5.4499	5.4643	5.3480
6	5.078	5.1594	5.1730	5.5848	5.5682	5.5230	5.4264

Table 1: Natural frequencies of a CCCC plate.

where ρ is the material density, G the shear modulus $\left(G = E/2(1+\nu)\right)$, E the modulus of elasticity and ν the Poisson's coefficient. Indices m and n are the vibration half-waves in axes x and y, and we use a shear correction factor $k = 0.8601$ Tab. 1 presents the natural frequencies of the CCCC plate, along with the frequencies of the three damages scenarios. We show the details of the damages cases in Tab. 2. Starting with single damage, then double damage, and multiple damages with variable severity in the last case. In the first two cases, we considered the same levels of damage severity. These scenarios are chosen in this manner to create a challenge for the optimization algorithms regarding two variables, first the location, then the severity. Then finally, the last case challenges the combination of variables simultaneously, with the complexity of

multiple damages, which makes the last scenario the hardest from an inverse problem perspective. This creates a problem with multiple local optima due to the similarity of the frequencies corresponding to the possible variable combinations.

Case	1	2	3
Element (Severity %)	15 (15%)	23 (20%)	33 (10%)
Element (Severity %)	–	87 (20%)	37 (15%)
Element (Severity %)	–	–	62 (20%)
Element (Severity %)	–	–	78 (5%)

Table 2: Damage case.

Fig. 1 shows the frequency modes of the CCCC plate, along with the discretization considered for the study. The plate has 100 elements. Higher modes 3 and 4 show unsymmetrical behavior.
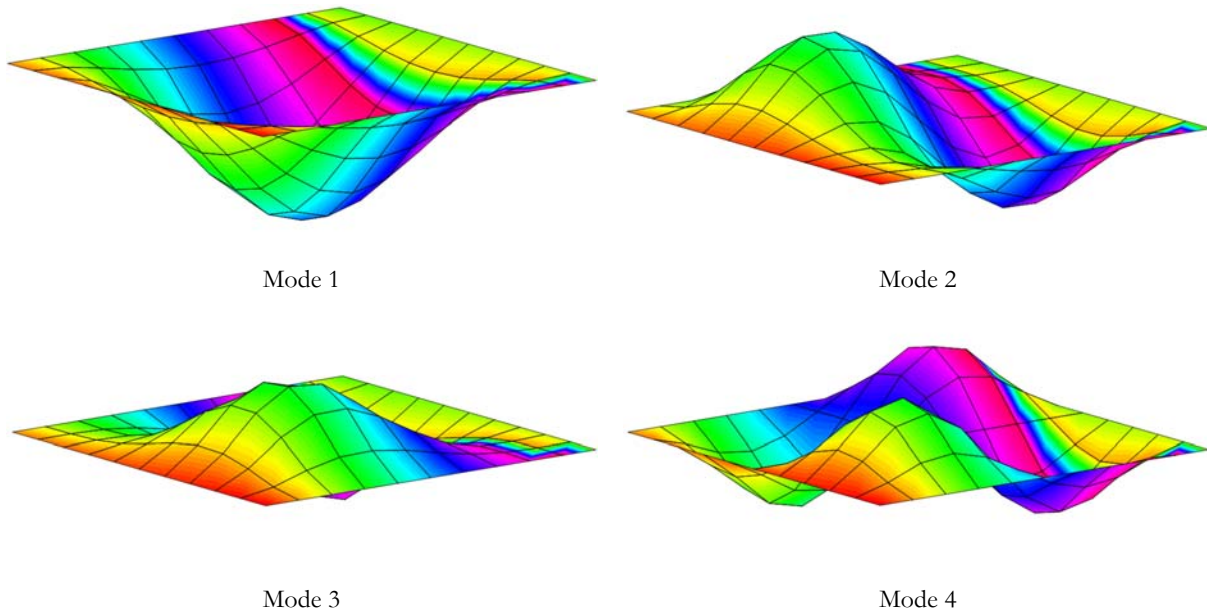
Mode 1

Mode 2

Mode 3

Mode 4

Figure 1: Modes of vibration for a CCCC plate.

Damage detection

The first case considers a single damaged element positioned near the side and on the centerline. Element number 15, coordinates (5,9). In this case, this element had 15% damage severity on the damage index. More details are shown in Fig. 2(a) and Fig. 2(b). Fig. 2(c) is the plot of Log(FRF) vs. the frequency in the case of healthy and damaged plates. It shows a shift in the natural frequencies, noticing a relatively large shift for the second natural frequency, with major variation in the last two natural frequencies.

The second case considers a double damaged plate; the two damaged elements are positioned near the side as well, closer to the plate corners, namely the element number 23 and the element number 78 with coordinates (3,8) and (7,2), respectively. In this case, both elements have 20% damage severity on the damage index. The details are shown in Fig. 3(a) and Fig. 3(b). Fig. 3(c) is the plot of Log(FRF) vs. the frequency in the case of healthy and damaged plates. It shows a slight shift in the natural frequencies in the first four natural frequencies, with significant variation in the last two natural frequencies, similar to the first case.

In the last case, the considered plate has four damaged elements, with two damages closer to the sides and two elements closer to the center of the plate, namely the element numbers 33, 37, 62, and 78, with coordinates (3,7), (2,4), (7,7) and (8,3) respectively. In this case, elements have different damage severity of 10%, 5%, 15%, and 20 %, respectively. The details are shown in Fig. 4(a) and Fig. 4(b). In this case, Fig. 4(c) shows a more significant shift in most natural frequencies with a

minor shift in the first and second peaks. We notice that multiple damages do not affect the first natural frequencies the same way the single damage case does. There is a similarity between the natural frequencies affected in double damages and multiple damages, with the difference in the magnitude of the shift.
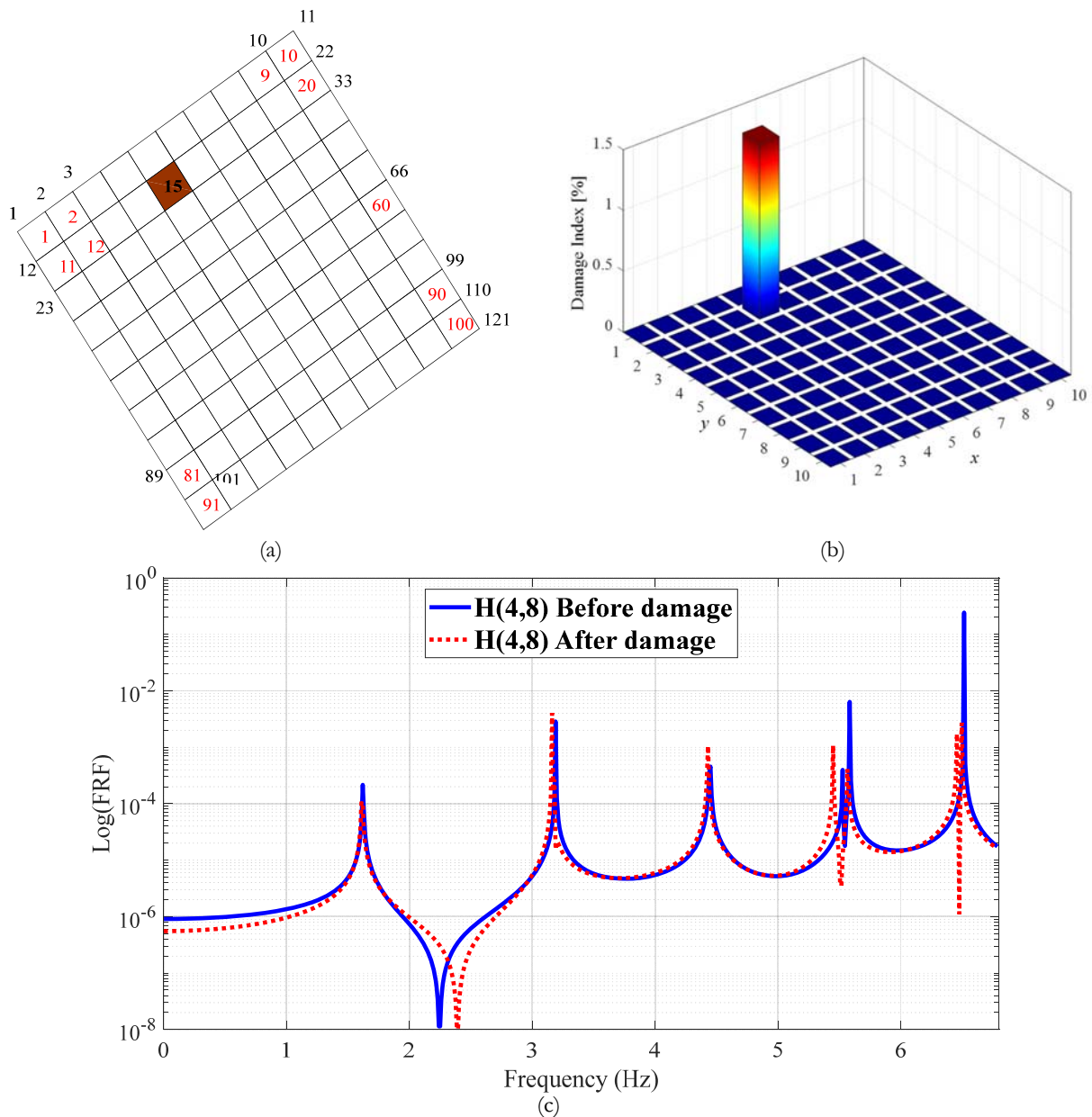

Figure 2: Damage case 1 – CCCC plate. (a) Element mesh numbers, (b) damage Index, and (c) Comparison of $H(4,8)$ before and after damage.

Optimization: Severity of damage detection

In this section, we compare the performance of damage identification results of three metaheuristic algorithms, Namely the Wild Horse Optimizer (WHO), Harris Hawks Optimization (HHO), and Arithmetic Optimization Algorithm (AOA), coupled with the Frequency Response Function (FRF) damage index. The fitness function is defined as the error between $\beta_i^{Measured}$ and $\beta_i^{Calculated}$, and is calculated from the following equation:

$$OF = \sum_{i=1}^{N} \left| \beta_i^{Measured} - \beta_i^{Calculated} \right| \qquad (26)$$

where $\beta_i^{Calculated}$ is extracted from Eqn. 5 For a fair comparison, we used a population of 50 solutions and a maximum number of iterations equal to 100. Fig. 5 depicts the fitness convergence for the three damages cases. The WHO algorithm presents an obvious better performance in all cases, converging toward a value of 1E-10 in the first and second case, 1E-5 in the last case. We also notice that the convergence speed is much higher in the case of single damage than in the cases where there are multiple damages. The case of four damages corresponds to the highest computational time. On the other hand, the HHO algorithm presents a good result in the single damage case, but it is outperformed in the other cases. Lastly, AOA algorithm presents the weakest performance on the three algorithms in these cases of damage detection.
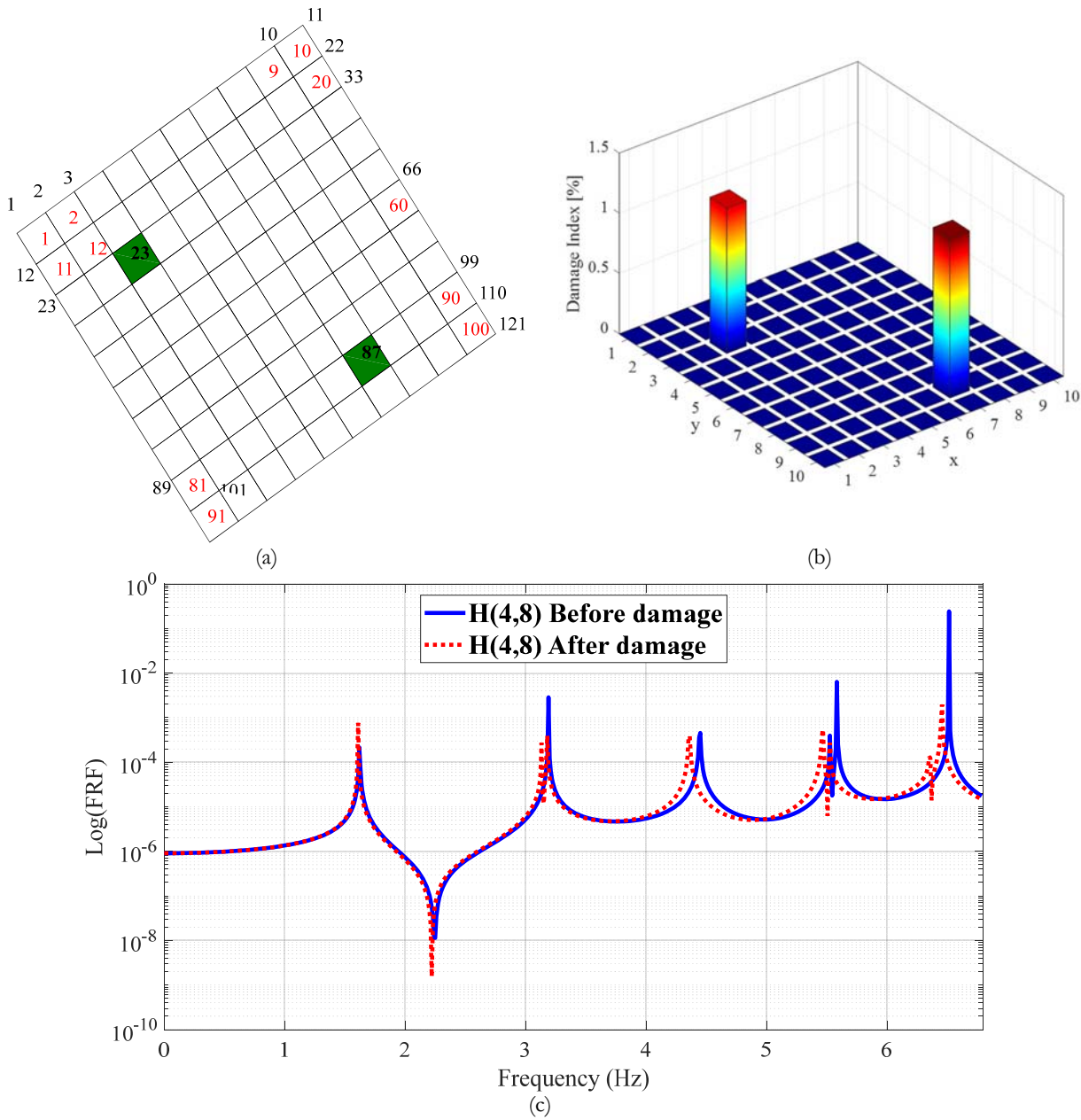

(a)

(b)

(c)

Figure 3: Damage case 2 – CCCC plate. (a) Element mesh numbers, (b) damage Index, and (c) Comparison of $H(4,8)$ before and after damage.
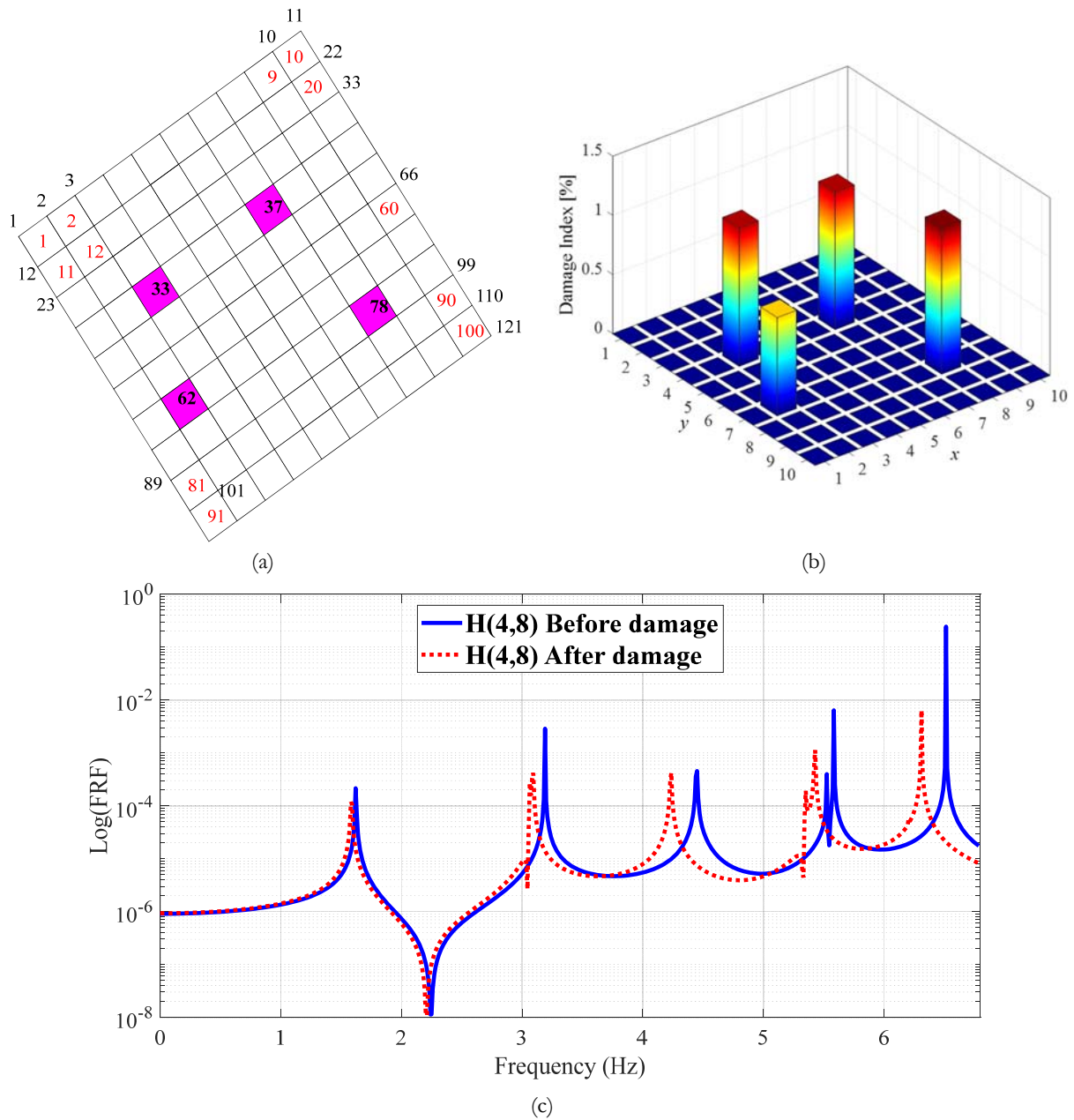
(a)

(b)

(c)

Figure 4: Damage case 3 – CCCC plate. (a) Element mesh numbers, (b) damage Index, and (c) Comparison of $H(4,8)$ before and after damage.

Tab. 3 shows the fitness values for all cases in some iteration pints. In this table, the performance of WHO algorithm is better from the earliest stage of the search, while the other two algorithms make progress but in a much slower manner. Fig. 6 presents the damage index convergence in the first case, comparing the performance of the three algorithms. This figure shows that all algorithms are efficient for this variable; as shown in Tab. 4, the three algorithms converge toward the actual damage severity, but the WHO algorithm accuracy is higher.
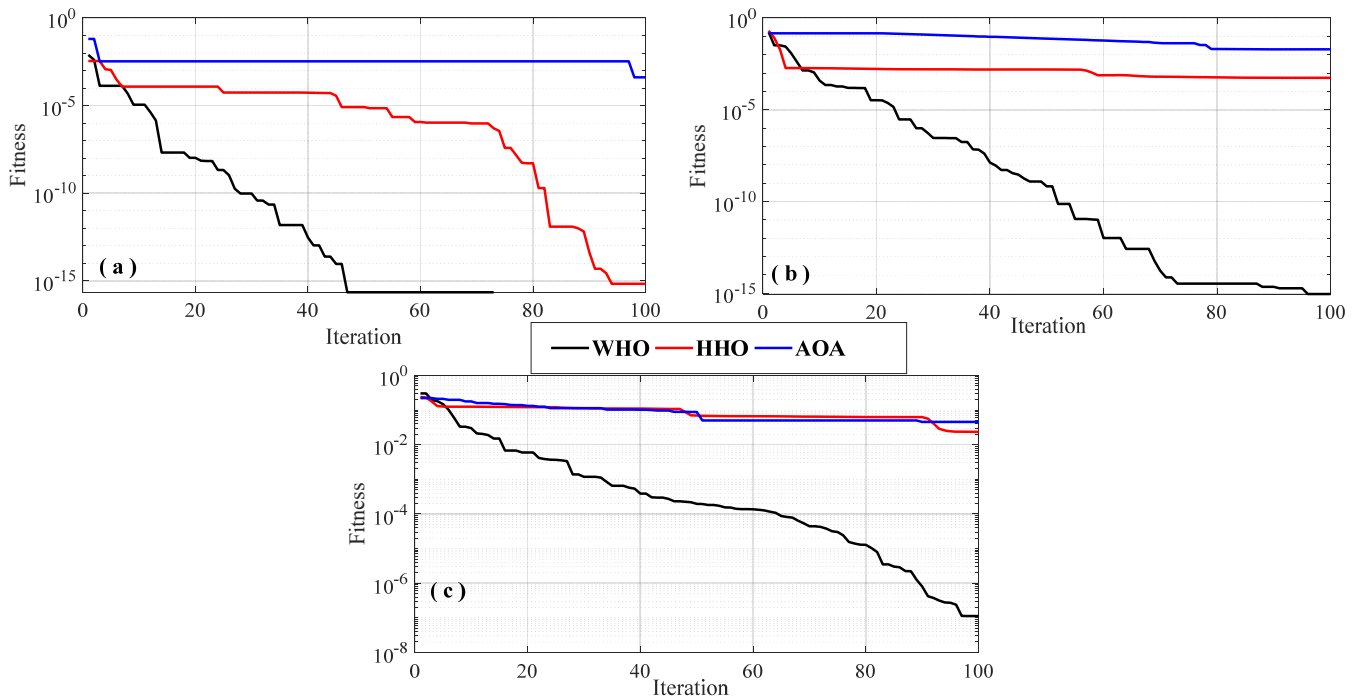
Figure 5: Fitness for all damage case – CCCC plate: (a) case 1, (b) case 2 and (c) case 3.

Iteration	Case 1			Case 2			Case 3		
	WHO	HHO	AOA	WHO	HHO	AOA	WHO	HHO	AOA
1	0.00403	0.003507	0.06316	0.032108	0.08396	0.14573	0.302640	0.229766	0.221604
10	1.12E-05	0.000120	0.00333	0.000387	0.00187	0.14573	0.029882	0.124371	0.178281
20	1.04E-08	0.000120	0.00333	3.32E-05	0.00173	0.14573	0.005921	0.121648	0.131715
30	9.63E-11	5.51E-05	0.00333	2.90E-07	0.00159	0.11978	0.001183	0.112177	0.113453
40	2.95E-13	5.51E-05	0.00333	1.36E-08	0.00157	0.09393	0.000386	0.109543	0.101677
50	2.22E-16	8.13E-06	0.00333	6.84E-10	0.00157	0.07503	0.000195	0.069667	0.087802
60	2.22E-16	1.17E-06	0.00333	1.03E-12	0.00077	0.05913	0.000136	0.066543	0.050043
70	2.22E-16	9.57E-07	0.00333	1.74E-14	0.00064	0.04168	4.41E-05	0.064518	0.050043
80	0	5.05E-09	0.00333	3.39E-15	0.00058	0.02033	1.28E-05	0.062890	0.050043
90	0	4.13E-14	0.00333	2.29E-15	0.00055	0.01958	8.08E-07	0.062193	0.045408
100	0	6.66E-16	0.00041	9.16E-16	0.00055	0.01958	1.13E-07	0.023583	0.045408

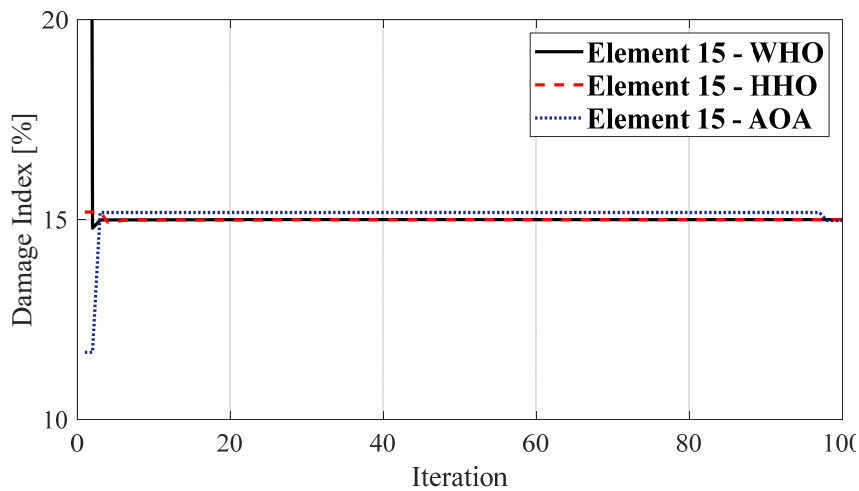Table 3: Variation of fitness for all damage cases – CCCC plate: case 3.

Figure 6: Damage Index – CCCC plate: case 1.

Iteration	Element 15		
	WHO	HHO	AOA
1	14.78	15.19	11.68
10	15.00	14.99	15.18
20	15.00	14.99	15.18
30	15.00	15.00	15.18
40	15	15.00	15.18
50	15	15.00	15.18
60	15	15.00	15.18
70	15	15.00	15.18
80	15	15.00	15.18
90	15	15	15.18
100	15	15	14.98
Actual	15%		

Table 4: Variation of damage Index – CCCC plate: case 1.

Fig. 7 presents the damage index identification in the second case, comparing the performance of the three algorithms as this case has two damages. Fig. 7(a) is dedicated to the severity of the damage in the first damaged element, and Fig. 7(b) is for the second element. This figure shows that all algorithms are efficient; as shown in Tab. 5, the three algorithms converge toward the actual damage severity. WHO algorithm accuracy is higher in this case as well, with noticeable fluctuation in the prediction of AOA algorithm; this is due to its exploitation technique; in this particular case, this method does not allow the algorithm to converge quickly.

The details in Tab. 5, suggest that the WHO algorithm performance is apparent in the early iterations. And the fluctuation in the AOA algorithm gets smaller by the progress of iterations. But it this algorithm is not able to find the actual damage severity with high accuracy within the 100 iteration limit.
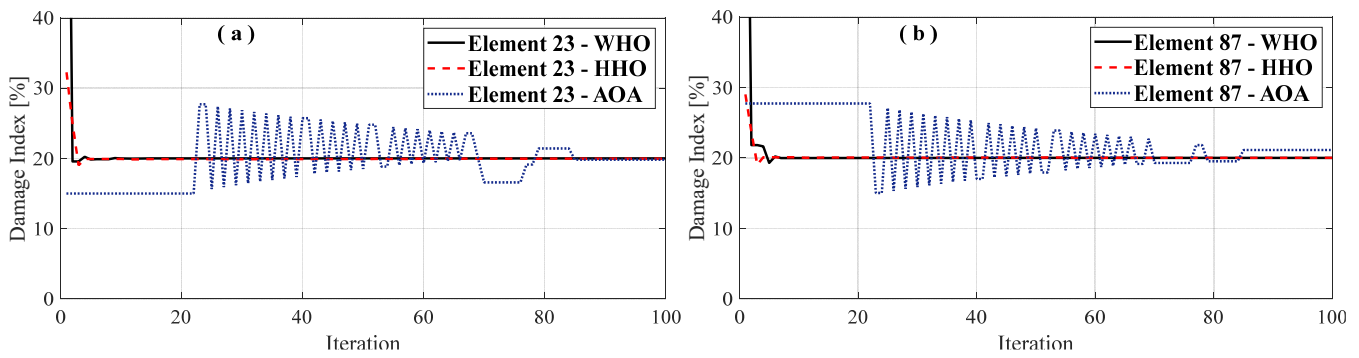
Figure 7: Damage Index – CCCC plate: case 2.

Iteration	Element 23			Element 87		
	WHO	HHO	AOA	WHO	HHO	AOA
1	19.57	24.88	15.00	21.83	23.78	27.73
10	19.98	19.93	15.00	19.98	20.09	27.73
20	20.00	19.90	15.00	20.00	20.06	27.73
30	20.00	19.89	26.87	20.00	20.04	15.87
40	20.00	19.90	25.72	20.00	20.05	17.02
50	20.00	19.90	18.51	20.00	20.05	24.23
60	20	19.95	19.31	20	20.02	23.42
70	20	19.96	16.58	20	20.02	19.28
80	20	19.96	21.42	20	20.02	19.53
90	20	19.96	19.82	20	20.01	21.12
100	20	19.96	19.82	20	20.01	21.12
Actual	20%			20%		

Table 5: Variation of damage Index – CCCC plate: case 2.

Fig. 8 presents the damage index identification in the last case, comparing the performance of the three algorithms of four damages. Fig. 8(a) is dedicated to the severity of the damage in the first damaged element, and Fig. 8(b) is for the second element, and Fig. 8(c) and Fig. 8(d) for the third and fourth elements, respectively. This figure shows that this case presents a challenge in the first 20 iterations; the WHO algorithm converges toward the correct damage severities in all four elements. The AOA and HHO algorithms are efficient in only two cases separately, with AOA algorithm corresponding to the highest error. The fluctuation technique is irregular in this case. HHO algorithm is noticed to have a less smooth convergence than the earlier cases.

Tab. 6 shows that the three algorithms converge toward the actual damage severity with different error levels. The WHO algorithm is the most accurate in this case, and the AOA algorithm corresponds to the most significant error rate.
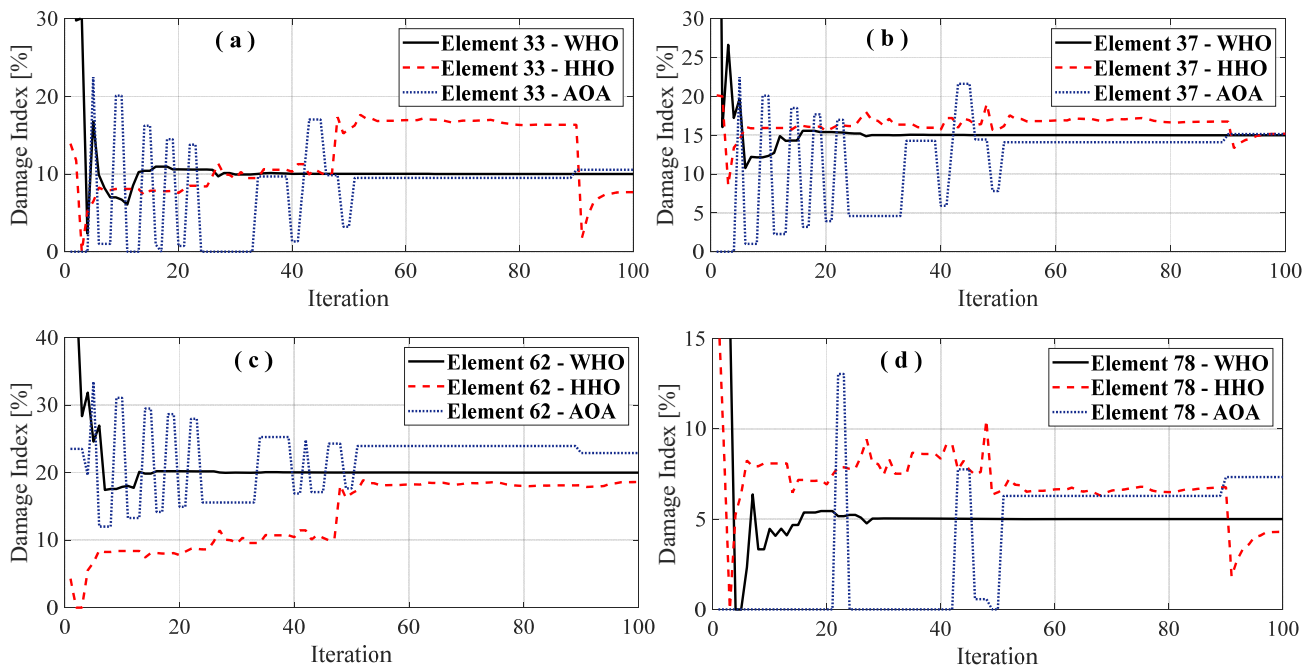
Figure 8: Damage Index – CCCC plate: case 3.

Iteration	Element 33			Element 37			Element 62			Element 78		
	WHO	HHO	AOA	WHO	HHO	AOA	WHO	HHO	AOA	WHO	HHO	AOA
1	29.77	11.76	0.00	16.05	20.02	0.00	47.71	0.00	23.52	17.88	8.68	0.00
10	6.70	8.08	20.09	12.34	15.93	20.09	17.87	8.39	31.08	4.46	8.08	0.00
20	10.60	7.56	0.75	15.41	15.54	3.95	20.19	7.78	14.93	5.44	6.91	0.00
30	9.93	9.39	0.00	15.02	16.29	4.59	19.99	9.48	15.58	5.04	7.48	0.00
40	10.02	10.29	1.33	15.03	15.71	5.92	20.03	10.44	16.91	5.02	8.34	0.00
50	10.03	15.80	3.21	15.01	15.74	7.81	20.02	16.92	17.65	5.01	6.48	0.00
60	10.02	16.94	9.50	15.01	16.90	14.09	20.01	18.23	23.94	5.01	6.65	6.29
70	10.00	16.87	9.50	15.00	17.04	14.09	20.00	18.41	23.94	5.00	6.58	6.29
80	10.00	16.32	9.50	15.00	16.65	14.09	20.00	18.02	23.94	5.00	6.50	6.29
90	10.00	16.35	10.54	15.00	16.76	15.14	20.00	18.12	22.90	5.00	6.77	7.33
100	10.00	7.67	10.54	15.00	15.18	15.14	20.00	18.59	22.90	5.00	4.30	7.33
Actual	10%			15%			20%			5%		

Table 6: Variation of damage Index – CCCC plate: case 3.

Fig. 9 presents the estimated severity of all damage cases compared to the actual severity level in the blue bar. The accuracy is highest for the first two cases. The considerable difference is shown in the case of four damages, shown in Fig. 9(c).
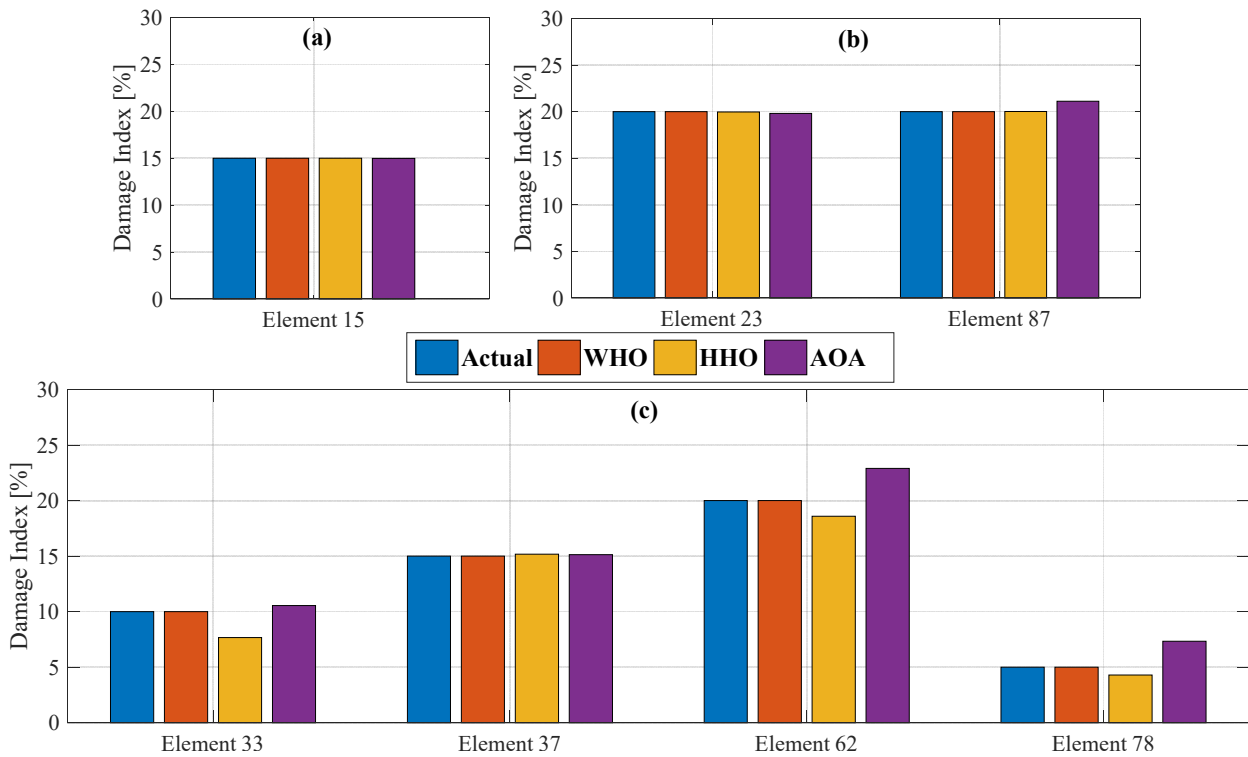

Figure 9: The estimated severity of all damage cases – CCCC plate.

Tab. 7. Compares the computations time in seconds, showing that the AOA and the WHO algorithm correspond to the lowest CPU time, with a slight advantage to the Who algorithm. HHO has the highest CPU time in all cases.

Case	WHO	HHO	AOA
1	4781.614611	11627.064433	4916.620988
2	4826.606647	11703.876201	4854.762505
3	4806.776610	11664.881398	4888.876639

Table 7: CPU Times.

CONCLUSION

In this paper, we studied the performance of three metaheuristic algorithms in the case of a CCCC plate; the problem is formulated as minimization of RFR damage indicator difference as an inverse problem. The plate is a square shape with 100 quadratic elements discretization, and the goal is to identify the damaged element and its damage severity level. The optimization algorithms showed acceptable performance within the considered search condition, with WHO algorithm corresponding to the best outcome in all cases and in terms of computational time. HHO and AOA algorithms presented relatively comparable performance, with AOA algorithm having an advantage in CPU time.

The single and double damage cases presented a moderate challenge compared to the third case, with four damages with different damage severity. Showing that the FRF indicator coupled with metaheuristic algorithms can easily handle the isolated variables, namely the element position and damage severity. While the case of multi variables with a higher number of damaged elements highlighted the good performance of WHO algorithm in the considerably limited number of iterations.

REFERENCES

[1] Dokeroglu, T., Sevinc, E., Kucukyilmaz, T., Cosar, A. (2019). A survey on new generation metaheuristic algorithms, Computers & Industrial Engineering, 137, pp. 106040.

[2] Gandomi, A.H., Yang, X.-S., Talatahari, S., Alavi, A.H. (2013). Metaheuristic algorithms in modeling and optimization, Metaheuristic applications in structures and infrastructures, pp. 1-24.

[3] Abdel-Basset, M., Abdel-Fatah, L., Sangaiah, A.K. (2018). Metaheuristic algorithms: A comprehensive review, Computational intelligence for multimedia big data on the cloud with engineering applications, pp. 185-231.

[4] Tiachacht, S., Bouazzouni, A., Khatir, S., Behtani, A., Zhou, Y.-L.-M., Wahab, M.A. (2018). Structural health monitoring of 3D frame structures using finite element modal analysis and genetic algorithm, Journal of Vibroengineering, 20(1), pp. 202-214.

[5] Zenzen, R., Belaidi, I., Khatir, S., Wahab, M.A. (2018). A damage identification technique for beam-like and truss structures based on FRF and Bat Algorithm, Comptes Rendus Mécanique, 346(12) pp. 1253-1266.

[6] Zenzen, R., Khatir, S., Belaidi, I., Le Thanh, C., Wahab, M.A. (2020). A modified transmissibility indicator and Artificial Neural Network for damage identification and quantification in laminated composite structures, Composite Structures, 248 pp. 112497.

[7] Cuong-Le, T., Nghia-Nguyen, T., Khatir, S., Trong-Nguyen, P., Mirjalili, S., Nguyen, K.D. (2021). An efficient approach for damage identification based on improved machine learning using PSO-SVM, Engineering with Computers, pp. 1-16.

[8] Khatir, S., Boutchicha, D., Le Thanh, C., Tran-Ngoc, H., Nguyen, T., Abdel-Wahab, M. (2020). Improved ANN technique combined with Jaya algorithm for crack identification in plates using XIGA and experimental analysis, Theoretical and Applied Fracture Mechanics, 107, pp. 102554.

[9] Chen, C., Yu, L. (2020). A hybrid ant lion optimizer with improved Nelder–Mead algorithm for structural damage detection by improving weighted trace lasso regularization, Advances in Structural Engineering, 23(3) pp. 468-484.

[10] Livani, M., Khaji, N., Zakian, P. (2018). Identification of multiple flaws in 2D structures using dynamic extended spectral finite element method with a universally enhanced meta-heuristic optimizer, Structural and multidisciplinary optimization, 57(2), pp. 605-623.

[11] Huang, M., Cheng, S., Zhang, H., Gul, M., Lu, H. (2019). Structural damage identification under temperature variations based on PSO–CS hybrid algorithm, International Journal of Structural Stability and Dynamics, 19(11) pp. 1950139.

[12] Benaissa, B., Hocine, N.A., Belaidi, I., Hamrani, A., Pettarin, V. (2016). Crack identification using model reduction based on proper orthogonal decomposition coupled with radial basis functions, Structural and Multidisciplinary Optimization, pp. 1-10.

[13] Benaissa, B., Köppen, M., Wahab, M.A., Khatir, S. (2017). Application of proper orthogonal decomposition and radial basis functions for crack size estimation using particle swarm optimization. in Journal of Physics: Conference Series. : IOP Publishing 012014.

[14] Samir, K., Brahim, B., Capozucca, R., Wahab, M.A. (2018). Damage detection in CFRP composite beams based on vibration analysis using proper orthogonal decomposition method with radial basis functions and cuckoo search algorithm, Composite Structures, 187, pp. 344-353.

[15] Mishra, M., Barman, S.K., Maity, D., Maiti, D.K. (2020). Performance studies of 10 metaheuristic techniques in determination of damages for large-scale spatial trusses from changes in vibration responses, Journal of Computing in Civil Engineering, 34(2), pp. 04019052.

[16] Ding, Z., Li, J., Hao, H.(2020). Non-probabilistic method to consider uncertainties in structural damage identification based on Hybrid Jaya and Tree Seeds Algorithm, Engineering Structures, 220, pp. 110925.

[17] Moezi, S.A., Zakeri, E., Zare, A. (2018). A generally modified cuckoo optimization algorithm for crack detection in cantilever Euler-Bernoulli beams, Precision Engineering, 52, pp. 227-241.

[18] Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., Gandomi, A.H. (2021). The Arithmetic Optimization Algorithm, Computer Methods in Applied Mechanics and Engineering, 376, pp. 113609. DOI: 10.1016/j.cma.2020.113609.

[19] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H. (2019). Harris hawks optimization: Algorithm and applications, Future Generation Computer Systems, 97, pp. 849-872. DOI: 10.1016/j.future.2019.02.028.

[20] Naruei, I., Keynia, F. (2021). Wild horse optimizer: a new meta-heuristic algorithm for solving engineering optimization problems, Engineering with Computers, DOI:10.1007/s00366-021-01438-z.

[21] Wang, Z., Lin, R., Lim, M. (1997). Structural damage detection using measured FRF data, Computer methods in applied mechanics and engineering, 147(1-2), pp. 187-197.

[22] Khatir, S., Tiachacht, S., Le Thanh, C., Ghandourah, E., Mirjalili, S., Abdel Wahab, M. (2021). An improved Artificial Neural Network using Arithmetic Optimization Algorithm for damage assessment in FGM composite plates, Compos. Struct., 273, pp. 114287. DOI: 10.1016/j.compstruct.2021.114287.

[23] Dawe, D.J., Roufaeil, O.L. (1980). Rayleigh-Ritz vibration analysis of Mindlin plates, J. Sound Vibrat., 69(3), pp. 345-359. DOI: 10.1016/0022-460X(80)90477-0.

[24] Liew, K.M., Wang, J., Ng, T.Y., Tan, M.J. (2004). Free vibration and buckling analyses of shear-deformable plates based on FSDT meshfree method, J. Sound Vibrat., 276(3), pp. 997-1017. DOI: 10.1016/j.jsv.2003.08.026.

[25] Ferreira, A.J. (2009). MATLAB codes for finite element analysis. Springer.