OsloMet – Oslo Metropolitan University

Department of Civil Engineering & Energy Technology
Section of Civil Engineering

ACCESSIBILITY - Open

Master Program in Structural Engineering & Building Technology

MASTER THESIS

TITLE OF REPORT	DATE
Structural damage localization and quantification in plane truss structures using optimization techniques	25.05.2023
	PAGES / ATTACHMENTS 90/0
AUTHOR(S) Merhawi Misgina Bayre	SUPERVISOR(S) Vagelis Plevris Alejandro Jimenez Rios

SUMMARY / SYNOPSIS

This thesis addresses the problem of identifying structural damage by employing various modal correlation criteria. The identification of structural damage is approached as an optimization problem, which is solved using two optimization techniques: Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). These techniques aim to achieve optimal results in detecting damage. The optimization process utilizes objective functions that rely on dynamic analysis data, including modal flexibilities, natural frequencies, and mode shapes of the structure.

KEYWORDS
Structural Damage Identification
Optimization
Natural Frequencies

Acknowledments

This thesis is the concluding work for the Master Program in Structural Engineering and Building Technology at Oslo Metropolitan University. This work is conducted within a period of a half a year, from January to May 2023, and counts for a total of 30 ECTS credits.

My sincere appreciation goes out to my supervisors Adjunct Professor Vagelis Plevris and Postdoctoral Fellow Alejadro Jimenez Rios, who provided encouragement, assistance, and advice for my thesis. I am very appreciative of the invaluable knowledge and experience that I have gained while working under their direction, and I am very grateful for their continuous support. They helped me initially with establishing the topic and objective of the thesis. They additionally directed me in the route that this thesis has followed throughout our frequent sessions and constructive feedback.

Abstract

Civil infrastructures are susceptible to threats from both nature and human activity; as they are built and used, they deteriorate, potentially resulting in structural damage or even collapse. The detection of structural damage is an important field of study that aims to identify and quantify any possible damage to structures such as bridges, buildings, and other infrastructure. Early detection of structural deterioration benefits the identification of cracks, flaws, and other possible safety issues in civil infrastructure. Identifying and quantifying structural damage with methods based on dynamic analysis data of structures is the main objective of the present study.

The damage identification problem is approached as an optimization problem, which is solved using two optimization techniques: Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). Three objective functions based on dynamic analysis data of the structures such as modal flexibilities, natural frequencies and mode shapes are used in the optimization process. This data was gathered by developing a program that performs the dynamic analysis of structures using the Finite Element Method (FEM). The effectiveness of each objective function is assessed through evaluations conducted on three damage scenarios involving a 10-bar truss structure. The impacts of noise and damage levels on damage detection are investigated.

Table of Contents

List of figures

List of Tables

List of abbreviations

PSO	-	Particle Swarm Optimization
GA	-	Genetic Algorithm
FEM	-	Finite Element Method
NDT&E	-	Non-destructive & Evaluation Techniques
MAC	-	Modal Assurance Criterion
COMAC	-	Co-ordinate Moal Assurance Criterion
DOF	-	Degree of freedom
MTMAC	-	Modified Total Modal Assurance Criterion
MACFLEX	-	Modal Flexibility Assurance Criterion
SQP	-	Sequential Quadratic Programming (SQP)
QP	-	Quadratic Programming
NLP	-	Nonlinear Programming
GRG	-	Generalized Reduced Gradient
FEA	-	Finite Element Analysis

1. Introduction

Civil infrastructures are susceptible to threats from both nature and human activity; as they are built and used, they deteriorate, potentially resulting in structural damage or even collapse. Structural damage will increase the possibility that the structure won't be able to operate for the remainder of its designed lifespan. Therefore, it is essential to identify structural damage early to reduce the risk of sudden collapse and increase the stability and dependability of buildings.

The detection of structural damage is an important field of study that aims to identify and quantify any possible damage to structures such as bridges, buildings, and other infrastructure. Early detection of structural deterioration benefits in the identification of cracks, flaws, and other possible safety issues in civil infrastructure. Identifying and fixing these problems could potentially avoid catastrophic failures that jeopardize human lives. It can also assist in prolonging the life of existing infrastructure, lessening the need for new construction, which can have a substantial environmental impact. Developing novel structural damage detection methods and technologies may lead to new research, innovations, and applications that can increase our knowledge of materials, structures, and their behavior.

In this field, vibration-based technologies are often used since they enable the non-invasive, non-destructive study of structures. Researchers may utilize changes in modal characteristics, namely natural frequencies, and mode shapes, which can be used to detect and locate any damage or faults, by monitoring the vibration of a structure. In recent years, these modal parameters have gained widespread use in damage-detection techniques and can be readily and inexpensively extracted from measured vibration responses [1].

Several pieces of literature studied and gathered early vibration-based damage detection approaches. Doebling et al.[2], for example, exhaustively examined vibration-based damage detection approaches and their application to diverse structures. Sohn et al. [3] offered an overview of structural health monitoring studies that have arisen in technical literature. Hou and Xia [4] delivered an in-depth discussion of the most recent breakthroughs in vibration-based damage diagnostics for civil engineering structures.

This thesis addresses the problem of identifying structural damage by employing various modal correlation criteria. The identification of structural damage is approached as an optimization problem, which is solved using two optimization techniques: Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). These techniques aim to achieve optimal results in detecting damage. The optimization process utilizes objective functions that rely on dynamic analysis data, including modal flexibilities, natural frequencies, and mode shapes of the structure. To acquire the necessary dynamic analysis data, software is developed that performs dynamic analysis using the Finite Element Method (FEM). The effectiveness of each objective function is assessed through evaluations conducted on multiple damage scenarios involving a 10-bar truss structure.

1.1. Justification

Several of the Sustainable Development Goals of the United Nations have a close connection to the employment of vibration-based technologies for structural damage identification. For instance, Goal 9 strives to build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation [5]. We are able to make sure that buildings are safe, functional, and sustainable for an extended period of time by identifying and repairing damage in structures. Additionally, we can reduce the environmental impact of our infrastructure maintenance and repair efforts by using non-invasive, non-destructive techniques like vibration analysis. This will help us reach Goal 13, which asks for immediate action to combat climate change and its impacts [6]. Furthermore, by encouraging the construction of more durable and robust structures and infrastructure, the adoption of vibration-based structural damage diagnosis tools may support the EU's long-term strategic plan for 2050 [7]. Structures may be made safe and long-lasting, which will lessen their carbon footprint and support sustainable development.

Overall, the use of vibration techniques for structural damage detection is a crucial instrument for advancing sustainable infrastructure and guaranteeing the security and well-being of populations all over the globe.

1.2. Objectives

Determining the degree and location of structural damage is the objective of this paper. This is accomplished by making use of certain unique and readily measurable structural dynamics. Two optimization methods, PSO and GA, are employed to address this damage detection problem. In this unconstrained optimization problem, the values of certain dynamic features are compared between the experimental and numerical models of the structure using objective functions that need to be minimized.

1.3. Thesis outline

To get an overview of how the paper is structured and what is included in each chapter, a brief summary of the content is described in this section. Chapter 2– The literature review is presented. Here, the theoretical backgrounds of damage identification methods and well-known optimization algorithms are presented and explained. Relevant literature was found with the help of the academic search engine, Scopus. Chapter 3, – Using numerical examples, this chapter presents the application of the optimization algorithms and the different modal correlation criteria discussed in the literature review. The aim of this chapter is to familiarize the author with these methodologies and demonstrate them with numerical examples for better understanding, which will subsequently be employed in the case study of this paper. Chapter 4 – Presents a short introduction of the case study and the aim of the paper. Chapter 5, Introduces the methodology used for the case study. Chapter 6- Presents the numerical example, results and discussion of the results. Chapter 7 – Concluding remarks and the possibility of future work.

2. Literature review

2.1. Structural damage identification

In order to acquire information about the condition of structures using observed dynamic characteristics, NDT&E in the computational mechanics industry has shown considerable interest in vibration-based structural damage detection. These methods make use of structural vibration properties such as modal domains, frequency response functions, and natural frequencies, which may be used to identify the presence, location, and severity of structural damage [8]. Here, it is anticipated that modifications to a structure's natural frequencies, frequency response functions, and mode shapes would provide dynamic data that varies from that of the healthy (undamaged) system. Modal parameters are dependent on the physical properties of the structure, specifically stiffness and mass. By utilizing tools like FEM, we can detect and locate damaged elements in a structure through the update of modal parameters. This approach may be utilized, even in large-scale structures, to determine the degree and location of damage.

The term "damage detection" has a wide range of applications and is not typically limited to constructions. This demonstrates that the effects of diverse approaches vary significantly. However, in this study, the phrase will relate to structures [9].

When discussing structural damage identification, we have 4 different levels [9]: (Level 1) Detection, (Level 2) Localization, (Level 3) Assessment, and (Level 4) Consequence. Level 1 provides us with a qualitative indication of the structure’s deterioration. Information about the damage’s most probable location is included in Level 2. Level 3 includes information on the severity of the damage and is provided at level 3. Finally, the fourth level (Level 4 - Consequence) includes information concerning the structure's safety, given a specific damage condition [10].

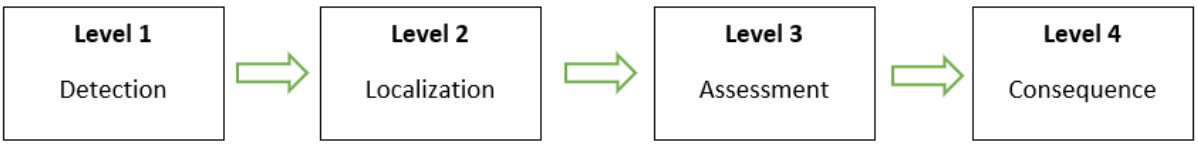

Figure 1 Levels of structural damage identification

It is common practice to transform the task of detecting structural damage into a single-objective optimization problem with or without constraints. The aim of this optimization problem is to identify a suitable objective function that best reflects the update of certain modal parameters. This function must have the modal parameters necessary for accurately identifying even little structural damage, and its estimate must be simple. A wide range of various objective functions for damage detection problems can be found and will be detailed in the chapters that follow.

2.2. Damage identification model

The stiffness and mass of a structure and its dynamic features are quantitively affected by structural damage if it is correctly modeled using FEM. The global mass matrix is shown to be unchanged in both the healthy and damaged structures. For the vast majority of practical applications, this assumption is thought to be quite correct. The following equation is generally known as the eigenvalue for a healthy (undamaged) structure [11]:

$$\left([K] - \omega_j^2[M]\right)\{\varphi_j\} = 0, j = 1,2, \ldots\ldots, N_m \quad (1)$$

Where:

[K] – The global stiffness matrix.

[M] – The global mass matrix.

$\{\varphi_j\}$ – Vibration mode shape vector.

ω_j – Natural frequency corresponding to j-th vibration mode shape $\{\varphi_j\}$.

N_m – The total number of vibration mode shapes obtained.

And for a damaged structure, Equation (1) becomes:

$$\left([K_d] - (\omega_j^d)^2\,[M]\right)\{\varphi_j{}^d\} = 0, j = 1,2, \ldots\ldots, N_m \quad (2)$$

Where:

$[K_d]$ – The damaged global stiffness matrix.

[M] – The global mass matrix.

$\{\varphi_j{}^d\}$ – Vibration mode shape vector in the damaged structure.

$\omega_j{}^d$ – Natural frequency corresponding to j-th vibration mode shape $\{\varphi_j{}^d\}$.

N_m – The total number of vibration mode shapes obtained.

2.3. Vibration-based structural damage identification methods

A wide variety of approaches for structural damage identification based on changes in dynamic characteristics are presented below. Additionally, different modal correlation criteria used in structural damage identification are presented.

2.3.1. Mode shape-based methods

One of the crucial strategies developed to solve the shortcomings of approaches like natural frequencies was the mode shape approach, which was used by Chen and Garba [12] in the early 1990s. A collection of modal characteristics of a structure may be calculated from its vibration response data, as shown by a variety of literature in modal analysis, system identification, and health monitoring. The identification of damage, may then be done using these modal parameters. The dynamic properties of a structure, such as modal parameters, may also be predicted by an accurate and verified finite element (FE) model of the structure. By utilizing them to directly estimate the mass and flexibility matrices of the structural model, for instance, or by directly comparing them with the equivalent one produced from an initial analytical model of the structure. Direct comparison of experimental and analytical modal characteristics has shown that when measuring structural damage, comparing mode shapes rather than natural frequencies provides a more accurate indicator of the damage to the structure [13].

Mathematical formulae are often used to compare two mode shapes that were received from two different sources, such as an undamaged model and a model that had been damaged but still corresponded to the same mode of vibration. To measure the consistency and correlation of estimations of the modal vector, two useful mathematical tools are the Co-ordinate Modal Assurance Criterion (CoMAC) and the Modal Assurance Criterion (MAC) [14].

The Modal Assurance Criterion (MAC)

The most commonly used technique for assessing differences in mode shapes between the undamaged and damaged models is the modal assurance criterion (MAC). This technique takes value between zero, representing no consistent correspondence between the models and one, representing a consistent correspondence of the two models, and is dependent on the mode shapes. If the two mode shapes are consistent, i.e., they differ only by a scalar factor of proportionality, MAC takes a value of one. However, MAC takes a value of zero, if the two mode shapes are orthogonal to each other [15] [16]. MAC is defined as in Equation (3):

$$MAC_{(i,j)} = \frac{\left|\{\varphi_A\}_i^T \ \{\varphi_B\}_j\right|^2}{\left(\{\varphi_A\}_i^T \ \{\varphi_A\}_i\right) \ \left(\{\varphi_B\}_j^T \{\varphi_B\}_j\right)} \quad (3)$$

Here $\{\varphi^A\}_i$ represents the structure's undamaged state's normalized i-th mode shape vector ,and $\{\varphi^B\}_k$ represents the damaged state's normalized k-th mode shape vector.

As previously stated, the modal vectors are not consistent if the modal assurance criteria has a value of zero. This can be due to the following reasons [16]:

- There is movement in the system. Any time the system experiences a change in mass or stiffness while being tested, this might happen.
- It is a nonlinear system. Distinct exciter placements or excitation signals will result in distinct frequency response functions that exhibit different system nonlinearities. Additionally, the various nonlinear properties will not be handled consistently by the modal parameter estimation algorithms.

The modal vectors are consistent if MAC is close to unity, which is an indication. This does not imply that they are always right. The modal assurance criteria does not indicate this if the same mistakes, random or bias, are present in all modal vector estimations.

The Co-ordinate Modal Assurance Criterion (CoMAC)

The effect of individual DOF's on vector similarity is one of the topics of interest in the comparison of two sets of modal vectors. The Coordinate Modal Assurance Criterion (COMAC), an extension of the MAC created by Lieven and Ewins [17], is a widely used criterion for comparing two sets of mode shapes which are obtained from different structures at a certain degree of freedom (DOF). The use of the criterion requires two steps of computation. The modes from the two sets are matched using the MAC in the first step. After creating the set of NM mode pairings that will be correlated, the COMAC's second step involves calculating the correlation values at each coordinate, across all the correlated pairs [17]. The COMAC factor is defined as follows for a structure's coordinate *k* and *n* mode shapes [18]:

$$COMAC_k = \frac{\left(\sum_{i=1}^{n}\left|\varphi_k^{(i)}\,\psi_k^{(i)}\right|\right)^2}{\sum_{i=1}^{n}\left[\left(\varphi_k^{(i)}\right)^2\right]\cdot\sum_{i=1}^{n}\left[\left(\psi_k^{(i)}\right)^2\right]} \tag{4}$$

The COMAC will take a value between 0 and 1 for a certain DOF. It gives us a global index that compares all of the comparable mode shapes from the two separate sets at the given DOF. In contrast, a high value COMAC implies that the two sets of mode shapes at that DOF are concordant. A low COMAC value shows that the two sets of mode shapes at the given DOF are inconsistent or dissimilar to one another.

2.3.2. Natural frequencies and mode shapes

Compared to other dynamic properties, the structure's natural frequency is the most straightforward and precisely determined. The natural frequency-based approach therefore becomes one of the most popular methods for identifying damage. The mass, rigidity, and other structural attributes of the visible structure are reduced by damage, which lowers the natural frequency value. One of the early proposals for using natural frequencies for damage detection was made in the 1970s [19]. This research included experimental testing on variety of components, such as straight prismatic bars, a doubly tapered bar, and an automotive camshaft. The findings were consistent regarding the locations and sizes of the anticipated and actual damage sites. The Modified Total Assurance Criterion (MTMAC) falls under this category, as it is dependent on mode shapes and frequencies.

The Modified Total Assurance Criterion (MTMAC)

The MAC criteria has a drawback in that it only considers the eigenvectors and ignores the eigenvalues of the different mode shapes of the structures. This indicates that the MAC criteria won't be able to identify any change if there is uniform damage. With a uniform damage, the structure here is more flexible (the eigenperiod is longer), but the eigenvectors are unaffected and continue to be the same as previously. Natural frequencies provide a comprehensive picture of the structure and may be properly located via dynamic measurements [20].

MTMAC is an effective objective function that combines MAC and natural frequencies. The MTMAC is defined as follows [20]:

$$MTMAC_j = \frac{MAC_i}{1 + \left| \frac{\omega_{A(i)}^2 - \omega_{B(i)}^2}{\omega_{A(i)}^2 + \omega_{B(i)}^2} \right|} \tag{5}$$

Where MAC_j is defined as in equation (3), $\omega_{A(i)}$ and $\omega_{B(i)}$ are the natural frequencies corresponding to the j-th mode. The MTMAC is a vector with the same number of values as the mode shapes taken into consideration in this fashion. In order to take into account, the entire number of mode forms (N_m), a total criterion is defined as follows [21]:

$$MTMAC = \prod_{j=1}^{N_m} MTMAC_{(j)} \tag{6}$$

A score close to one indicates an almost perfect correlation between the undamaged and damaged numerical findings, while an index equal to zero denotes no connection between the two sets of mode shapes and natural frequencies.

2.3.3. Modal flexibility

Modal flexibility captures the effect of both natural frequencies and mode shapes. The contributions from all accessible mode shapes and associated natural frequencies are accumulated to form it. Damage has been shown to have an effect on a structure's stiffness matrix, specifically reducing the stiffness of the individual damaged elements. Increased structural flexibility results from a decrease in stiffness. It is desirable to employ changes in flexibility as an indication of damage in structural health monitoring rather than stiffness perturbation. This is because of the following aspects [22]:

- Since the lower modes dominate the flexibility matrix, accurate estimates may be made even with a limited number of lower modes.
- The modes and mode shapes produced by the system identification procedure immediately lead to the flexibility matrices.
- Iterative algorithms usually converge the fastest to high eigenvalues.
- These eigenvalues match the predominant low-frequency structural vibration components in flexibility-based approaches.

Therefore, the detected modal parameters may be used to easily generate the dynamically measured flexibility matrix, which can then be used as a damage detection approach. A position *i* on the structure's related modal flexibility may be shown by [23]:

$$F_i = \sum_{r=1}^{n} \frac{1}{\omega_r^2} \Phi_{ir} \Phi_{ir}^T \tag{7}$$

Where ϕ_{ir} is the magnitude of mass normalized modal vector at location *i* for mode *r*, ϕ_{ir}^T is the modal vector that has been transposed at location i for mode r, ω_r is the corresponding circular natural frequency to mode *r*, and *n* is the number of modes taken into consideration.

Or in matrix form:

$$[F] = [\boldsymbol{\Phi}] \cdot [\boldsymbol{\Lambda}]^{-1} \cdot [\boldsymbol{\Phi}]^{T} \tag{8}$$

Where:

$[\boldsymbol{\Phi}]$, $[\boldsymbol{\Phi}]^T$ – The mode shape matrix and its transpose, respectively.

$[\boldsymbol{\Lambda}]$ – The spectral matrix containing the eigenfrequencies of n vibrating modes, diagonal (ω_j^2).

The Modal Flexibility Assurance Criterion (MACFLEX)

The values of the flexibility matrices of two structures—one that has been damaged and one that has not—are compared using the modal flexibility assurance criterion. The following formulas may be used to determine each component of the MACFLEX vector ([1×n]) [20]:

$$MACFLEX_i = \frac{(F_A^{(i)T} F_B^{(i)})^2}{(F_A^{(i)T} F_A^{(i)})(F_B^{(i)T} F_B^{(i)})} \quad (9)$$

Where $F_A^{(i)}$ and $F_B^{(i)}$ are the i-th column vectors ([n×1]) of the flexibility matrices F_A and F_B, which represent the flexibility of healthy and damaged structures, respectively. Since it compares the identical vectors of each set of flexibilities, this criterion solely makes use of the diagonal terms of the MAC matrix as stated in equation (3). Thus, MACFLEX is a vector with the same number of values as the vectors included in the flexibility matrices [20].

The MACFLEX scalar value is created by multiplying the *n* distinct values of the MACFLEX vector [21]:

$$MACFLEX = \prod_{i=1}^{n} MACFLEX_i \quad (10)$$

As previously indicated, a score near to one indicates essentially minimal change in flexibilities and, as a consequence, a perfect correlation between the undamaged and damaged findings. An index equal to zero, however, signifies there is no connection between the two sets of flexibilities.

2.4. Optimization

The process of choosing the best parameters from a range of accessible options that produce the greatest or lowest value of an objective function in accordance with a specified set of criteria is known as optimization or optimization design problems. Numerous quantitative fields, including computer science, engineering, economics, and operational research, encounter optimization issues. Engineering and mathematics have long been interested in developing strategies for solving optimization problems [18].

However, structural optimization methods are a crucial part of contemporary structural engineering. The process of designing a structure to achieve a set of objectives, such as being reliable, light, and efficient, while reducing its weight or cost is known as structural optimization. There are normally three phases in this process [18]:

- Defining the goal function, potential limitations, and design variables. The characteristics that may be changed in the design, such as the structure's form, size, and material, are referred to as design variables. The criteria that the structure must fulfill, such as strength, stability, and durability, are the potential limitations. An indicator of a structure's performance, such as its weight or price, is its objective function [24].
- Defining the optimization challenge and
- Solving the problem.

The topic of evolutionary computing (EC) has had significant progress in the optimization domain when it comes to selecting an appropriate method. The vast majority of these algorithms take their cues from natural occurrences and are built using a mix of many rules and randomness. These methods can handle non-continuous, non-convex, and highly nonlinear solution spaces for challenging optimization problems [25].

Two significant groups of optimization algorithms are mathematical and metaheuristic ones. Mathematical algorithms are exact, predictable, and dependent on mathematical concepts, yet they may not succeed if the job is too difficult or large. Metaheuristic algorithms, on the other hand, describe methods for locating approximations of solutions. Instead of being a precise deterministic method, they are high-level techniques that direct the search in the direction of a desirable result [24]. Particle swarm optimization (PSO) and genetic algorithms (GA) are categorized as metaheuristic optimization techniques, while sequential quadratic programming (SQP) may be characterized as a mathematical optimization approach.

2.4.1. Optimization algorithms

This section introduces the fundamental ideas behind the optimization techniques employed in this study.

2.4.1.1. Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO), a metaheuristic optimization technique, was created by Kennedy and Eberhart in 1995 as a computer approach for optimizing continuous, nonlinear problems [26]. It is a probabilistic population-based approach, and the principles that govern the population are based on the behavior of bird flocks or bee swarms. Particles are the name given to the elements in the PSO algorithm. Each particle movement in the population is related to a subset of the population's particles and has an independent random component. This is also referred to as the particle's neighborhood and randomly produced velocity. In order to find a suitable solution to a problem, a collection of "Particles" is employed. Each particle's location is updated at each iteration based on its previous position and velocity, as well as the positions and velocities of the other particles in the swarm. The particles' movement is dictated by two types of information: the best position that the particle has achieved thus far, also known as the "personal best" position, and the best position that any particle in the swarm has achieved, otherwise known as the "global best" position. The algorithm alters the velocity of each particle as it moves towards its personal and global optimal positions at each time step [24]. PSO is a method that shows promise and has certain benefits over other optimization methods of a similar kind. This is mainly because the implementation is easier, there are less parameters that need to be changed, and the method uses less memory and computing power. PSO is also adaptable, making it simple to manage with goal functions [27].

The following is the updated equation for the particles' location and speed [25]:

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 \times (P_{best_i}{}^k - X_i^k) + c_2 r_2 \times (G_{best}{}^k - X_i^k) \quad (11)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (12)$$

Where,

V_i^k	velocity of particle i at iteration k
ω	inertia weight factor
c_1, c_2	acceleration coefficients
r_1, r_2	random numbers in the range [0, 1]
X_i^k	position of particle i at iteration k
$Pbest_i^k$	best position of particle i at until iteration k
$Gbest_i^k$	best position of the group i at until iteration k

2.4.1.2. Sequential Quadratic Programming (SQP)

Mathematical optimization is carried out using sequential quadratic programming (SQP). SQP has shown itself to be very successful in solving non-linear restricted models. For the solution of complex problems, it offers strong algorithmic tools and is based on a solid theoretical framework [28]. As is done for unconstrained optimization, the procedure closely parallels Newton's technique for restricted optimization. Using a Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton updating approach, the Hessian of the Lagrangian function is approximated at each iteration. As a result, a quadratic programming (QP) subproblem is created, the solution of which serves as the search direction for a line search technique. By resolving a series of QP approximations to the NLP, the SQP algorithm runs. The SQP uses the gradient data to solve a problem starting from a single search point. The objective function and restrictions must be continuously differentiable, as well as all other functions [27].

The generic structure optimization problem may be expressed mathematically as follows [29]:

$$min_x F_c(x) \tag{13}$$

Subject to:

$$h_i(x) = 0, \mathrm{i} = 1, \ldots\ldots, \mathrm{m}, \tag{14}$$

$$g_i(x) \leq 0, i = 1, \ldots\ldots, p \tag{15}$$

Where:

F_c is the objective function.

m is the number of equality constraints h(x)

p is the number of inequality constraints g(x)

x is the vector containing the design parameters

The preceding mathematical expression has an equality requirement. That is not often the case, for most structural optimization problems as they do not contain an equality constraint.

Given that the main objective of the SQP is to decompose the issue into a simpler subproblem, which may then be solved and used as the foundation for an iterative process. To be more precise, to retain the subproblem's linearity while taking into account the nonlinearities in the constraints. As the aim, the SQP technique employs a quadratic model of the Lagrangian function $\boldsymbol{\Lambda}$ [30].

$$\Lambda(x_k, u_i, v_i) = F_c(x_k) + \sum_{i=1}^{m} u_i h_i(x_k) + \sum_{i=1}^{p} v_i g_i(x_k) \tag{16}$$

The objective function at x_k is represented as Fc where u_i and v_i stand for the Lagrangian multipliers. The equality and inequality restrictions are hi and g_i, respectively. The QP subproblem may be generated by linearizing the equality and inequality requirements, and it has the following form:

$$min_{S_x}\left\{F_C(x_k) + \nabla F_C(x_k)^T S_x + \frac{1}{2} S_x^T He_k S_x\right\} \quad (17)$$

Subject to:

$$\nabla h_i(x_k)^T S_x + h_i(x_k) = 0, i = 1, \dots\dots, m. \quad (18)$$

$$\nabla g_i(x_k)^T S_x + g_i(x_k) \leq 0, i = 1, \dots\dots, p \quad (19)$$

Where T stands for the transposition operation, $S_x = x - x_k$ is the search direction, and x_k is the vector carrying the design parameter at iteration k. He_k is assumed to be the Lagrangian function's Hassian matrix at x_k.

The initial estimate for the manipulator chain solution is used to begin the numerical technique, and after each iteration k, the quadratic programming issue is resolved to provide a search direction S_x. For a particular choice of the step-length parameter ψ_k, the solution S_x may be utilized to produce a new iteration, x_{k+1}, as follows [27]:

$$x_{k+1} = x_k + \psi_k S_x \quad (20)$$

A fresh estimation of the Lagrangian multipliers is required to go on to the next iteration. Utilizing the quadratic sub-problem's optimum multipliers is a common strategy. Let's call these multipliers v_{qp} and u_{qp}. Thus, the following is how the updated multipliers u_{k+1} and v_{k+1} are obtained [29]:

$$x_{k+1} = x_k + \psi_k S_x \quad (21)$$

$$S_u = u_{qp} - u_k$$

$$v_{k+1} = v_k + \psi_k S_v \quad (22)$$

$$S_v = v_{qp} - v_k$$

In summary, a SQP approach determines the search direction using equations to solve the optimization issue given by the aforementioned equations (16) - (26) [29].

2.4.1.3. Genetic Algorithm (GA)

An iterative method based on random search with modifications to the search minimum in coordinate direction is known as a Genetic algorithm (GA). To avoid stopping at a local minimum, GA uses a probabilistic criterion, which may assist to account for little fluctuations in the objective function. It was first proposed by Holland in 1975 and popularized Goldberg in 1989 [29] [31], GA was motivated by biological natural evolution, and the population of candidate solutions goes through a process resembling natural selection and genetic variation.

Like PSO, GA begins its search with a population that is produced at random. The algorithm chooses parents at random from the existing population for each iteration, using them to generate offspring for the next generation. GA uses three basic sorts of procedures to assist it construct the following generation from the present one as part of the optimization process [32]:

I. Selection
II. Crossing
III. Mutation

'Selection', an operation that generates parents for the next generation, initiates the creation process. In biological populations, mating is represented by the 'crossover' operator. To conserve the attractive, enduring designs from the present to the future population, this operator serves as a filter. Finally, the ‘mutation' operator encourages population diversification and prevents the algorithm from becoming stuck in local optimums [33].

2.4.1.4. GRG-Nonlinear

There are three ways to solve problems in Excel using the Excel Solver addon: Simplex LP, Evolutionary, and GRG, which stands for "Generalized Reduced Gradient". GRG’s approach determines that it has arrived at an optimal solution when the partial derivatives equal zero by first examining the gradient or slope of the objective function when the input values change [34].

GRG is a method that has potential and is superior to the other nonlinear solution techniques in Excel, mainly because it is well-known for being quick. This speed does have a price, however [35].

The disadvantage is that the answer you get using this technique may not be the overall best one since it depends so much on the beginning conditions that the user sets. Due to the solver's propensity to stop at the local optimal value that is closest to the beginning circumstances, the solution you obtain may not be globally optimized in this case [35]. The function must also be smooth, and not include for example IF or ABS functions, in order for the GRG nonlinear solver to provide a suitable result.

The Solver Options window's GRG Nonlinear tab allows you to activate the GRG Multistart option, which is part of the GRG nonlinear method. This will make it possible for the conventional GRG nonlinear algorithm to generate a population of starting values that are spread at random and then assessed one by one. It is more likely that the global optimum will be reached when diverse beginning circumstances are tried several times.

2.4.1.5. Evolutionary

Due to the fact that the Evolutionary algorithm is more likely to discover a globally optimal solution than the GRG nonlinear algorithm, it is more resilient. The solver method's other drawback is that it is quite sluggish. The procedure is slow because it is based on the Theory of Natural Selection, which is effective in this situation since the ideal result has already been determined [36]. To put it simply, the solution begins with a random "population" of input value sets. These input value sets are entered into the model, and the outcomes are assessed in relation to the desired value [35].

A second population of "offspring" is produced by choosing the sets of input values that provide solutions that are most closely related to the desired value. The offspring are a "mutation" of the ideal set of initial population input values. The third population is then created once the second population has been examined and a winner selected. This continues until the objective function of one population relative to the next doesn't alter much beyond this point. This procedure takes a long time since each person in the population has to be assessed separately. Additionally, to identify the next optimum set of values, succeeding "generations" are randomly filled rather than utilizing derivatives and the slope of the objective function [35].

Excel may provide you with some control over the procedure to hasten the outcome by letting the user choose the Population Size and Mutation Rate. Reduced population size and or higher mutation rates may need even more populations in order to attain convergence, hence this has diminishing returns [35].

3. Investigation of the performance of various optimization algorithms and modal correlation criteria

This chapter introduces the implementation of the optimization algorithms as well as the various modal correlation criteria mentioned in the literature review, using numerical examples. The chapter is divided into two sub-chapters, where the first chapter exhibits examples using well-known optimization algorithms, and the second chapter exhibits the various criteria and how they perform.

The purpose of this investigative work is for the author to be familiar with these methods and exhibit using numerical examples for better comprehension. The aforementioned methods will later on be used in the case study of this thesis. The chosen methods are well-known within mathematics and engineering.

The numerical examples provided in this chapter are based on two research articles that were previously published. The first study, "A collection of 30 Multidimensional functions for global optimization benchmarking" by Plevris and Solórzano [37], investigated a total of 30 mathematical functions that can be used for optimization. The second paper, "Investigation of the performance of various modal correlation criteria in structural damage identification" by Georgioudakis and Plevris [21].

The numerical work for this chapter have been completed in Matlab and Excel.

3.1. Optimization algorithms

Finding the optimal element from a collection of potential options in terms of a particular criteria is the process of mathematical optimization. Engineering and mathematics have been interested in finding solutions to these optimization problems for centuries. Optimization difficulties are encountered in many quantitative fields, including engineering [37].

On a regular basis, new optimization techniques or fresh iterations of established ones are offered. Although there are certain known optimization techniques, when a variety of optimization problems are considered, no one technique beats all the others. A collection of well-known optimization problems is often chosen by the method's creators to test the algorithm on and serve as a benchmark for comparison with other, already-in-use methods. The benchmark functions, which are the objective functions selected for testing, are crucial in determining if the new suggested algorithm may be deemed successful when its performance is superior to or at least comparable to that of the current, well-established algorithms [37].

This section presents and explores in depth a set of ten mathematical functions that may be utilized for optimization. The functions are benchmark functions for unconstrained two-dimensional single-objective optimization problems are specified in two dimensions (D = 2). The ten objective functions used in this example are the Sphere, Ellipsoid, Sum of different

powers, Quintic, Drop_Wave, Alpine 1, Griewank's, Rastring's, HappyCat, and HGBat functions.

The performance of three optimization algorithms, namely Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Sequential Quadratic Programming (SQP), and three Excel Solver methods, namely GRG-nonlinear, GRG Multistart and Evolutionary are investigated. The implementation of the optimization algorithms is carried out in Matlab, while the Excel Solver methods are carried out using Excel.

Plevris and Solorzano [37] earlier looked at a study of a similar kind that included a collection of thirty mathematical functions that might be used for optimization. The functions can be used as benchmark functions for unrestricted multidimensional single-objective optimization problems since they are specified in multiple dimensions for any number of dimensions. In their research, they examined the effectiveness of two metaheuristic algorithms, GA and PSO, and a mathematical algorithm, SQP. The objective functions, optimization techniques, and applicability for each problem were the authors main focus. The authors also looked at how the performance of the optimizers and complexity of each challenge change as dimensionality increases.

The mathematical expressions of the ten functions used as optimization benchmark for this example are presented in Table 1.

Table 1 Mathematical expression of the 10 benchmark functions

No.	Function	Mathematical expression
1	Sphere	$x_1^2 + x_2^2$
2	Ellipsoid	$x_1^2 + 2x_2^2$
3	Sum of different powers	$\lvert x_1\rvert^2 + \lvert x_2\rvert^3$
4	Quintic	$\lvert x_1^5 - 3x_1^4 + 4x_1^3 + 2x_1^2 - 10x_1 - 4\rvert + \lvert x_2^5 - 3x_2^4 + 4x_2^3 + 2x_2^2 - 10x_2 - 4\rvert$
5	Drop Wave	$1 - \frac{1 + cos\,(12\sqrt{x_1^2 + x_2^2}}{0.5\,(x_1^2 + x_2^2) + 2}$
6	Alpine 1	$\lvert x_1 sin(x_1) + 0.1x_1\rvert + \lvert x_2 sin(x_2) + 0.1x_2\rvert$
7	Griewank's	$\frac{x_1^2 + x_2^2}{4000} - cos(x_1) \cdot cos(\frac{x_2}{\sqrt{2}}) + 1$
8	Rastring's	$x_1^2 + x_2^2 - 10cos(2\pi x_1) - 10cos(2\pi x_2) + 20$
9	HappyCat	$\lvert x_1^2 + x_2^2 - 2\rvert^{1/4} + 0.25(x_1^2 + x_2^2) + 0.5(x_1 + x_2) + 0.5$
10	HGBat	$\lvert (x_1^2 + x_2^2)^2 - (x_1 + x_2)^2\rvert^{1/2} + 0.25(x_1^2 + x_2^2) + 0.5(x_1 + x_2) + 0.5$

There is a specified minimum value since all the objective functions in this example correspond to minimization. The functions are investigated in two dimensions. For consistency, the minimum value for each function is set to zero. All the functions have no constraints and are considered unconstrained, but an upper and lower limit are set. The numerical work in this example is carried out both in Matlab and Excel Solver. The Matlab implementations for the optimization algorithms (GA, PSO, SQP) is executed using the Matlab commands *ga, particleswarm,* and *fmincon*, respectively.

Table 2 Input data for the various methods

Optimization Algorithm/ Excel Solver	**No. of Dimension/variables, D**	**Search range (*lb* & *ub*)**	**Implementations**	**Starting point values**
GA	2	[0, 100]	Matlab	-
PSO	2	[0, 100]	Matlab	-
SQP	2	[0, 100]	Matlab	[10; 8]
GRG-nonlinear	2	[0, 100]	Excel Solver	[10; 8]
GRG Multistart	2	[0, 100]	Excel Solver	-
Evolutionary	2	[0, 100]	Excel Solver	-

Table 2 presents the metaheuristic and mathematical optimizers, the number of dimensions for the mathematical functions, the search range (upper and lower bounds), implementation program and starting point values at the solution chosen when running the solving method. The lower and upper bounds set for all solving methods are 0 and 100 respectively. Notice that only SQP and GRG-nonlinear requires defining a starting point value. These values are chosen by the user, as these methods are dependable on initial values when searching for a solution. If the inserted starting point values are closer to the real solution, the faster will the solver finish solving the function. In the case of this example, since I have two design variables, two random values are chosen, 10 and 8 are set as initial values.

Table 3 Results for every method.

No.	Function	SQP	PSO	GA	GRG-Nonlinear	GRG-Nonlinear (Muli Start)	Evolutionary
1	Sphere	8.18E-07	0	0	0	0	0
2	Ellipsoid	9.86E-07	0	2.98E-08	0	0	0
3	Sum of different powers	2.67E-07	0	0	0	0	0
4	Quintic	4.1002	4	4.0005	8	4	0.0012
5	Drop-Wave	0.9753	0	0.638	0.968	0	0
6	Alpine 1	0.0602	0	1.91E-07	1.00E-04	0	0
7	Griewank	0.0666	0	0.106	0.0592	0	0
8	Rastring	6.47E-07	0	1.20E-06	163.1673	4.974	0
9	HappyCat	1	1	1	1	1	1
10	HGBat	1	0.5	1	0.5	0.5	0.5

Table 3 presents the results for each function using the optimization algorithms and Excel Solver methods mentioned above. Most of the results obtained using the various solving methods show a small variance or are in most cases similar. Every approach has its own distinct features, benefits, and downsides, just as every function has its own specific qualities. Functions such as HappyCat, HGBat and Griewank's are easily optimized by all solving methods, while others pose a real challenge to some.

The GRG-nonlinear shows the greatest variance of the results especially for the Quintic and Rastring's functions. The SQP also shows some variance, not as great as the GRG-nonlinear, but to some extent. This is mainly due to how these methods work. GRG-nonlinear and SQP require randomly chosen starting point values when solving a function, and when these values are far from the optimum solution, the solver methods tend to be stuck in a local minimum and result in a wrong solution rather than finding the global minimum value.

3.2. Modal correlation criteria examples

In this section, I will be focusing on investigating the performance of the different modal correlation criteria and how they are able to detect damage in structures via optimizing the correlation between the measured and predicted modal parametric change. Modal correlation criteria are used as a simple mathematical technique to assess the degree of consistency and correlation between estimates of modal vectors for the measured and predicted natural frequencies or mode shapes.

This example is based on prior research by Georgioudakis and Plevris [21], who evaluated the performance of several modal correlation criteria in structural damage detection. The structural damage detection problem is modeled as an optimization problem, with the differential evolution search technique used to solve it. A measure of consistency and correlation between estimates of modal vectors is shown by the objective functions utilized in the optimization process, which are based on various modal correlation criteria. By providing a variety of damage scenarios for a beam, they assessed the effectiveness of each of the objective functions.

The criteria that have been used in this example for the purpose of providing a measure of consistency and correlation between estimates of modal vectors are as follows:

- The Modal Assurance Criterion (MAC)
- The Modified Total Assurance Criterion (MTMAC)
- The Co-ordinate Modal Assurance Criterion (COMAC)
- The Modal Flexibility Assurance Criterion (MACFLEX)

Each criterion will be shown using a mathematical formulation. For better understanding and to demonstrate the various criteria employed, the numerical values for a given example as in the research article [21] of two structures will be replicated for this numerical example. Considered are the example structures A and B which correspond to the example which is examined in the research study. There are 9 active DOF's in each structure, and up to 4 eigenvalues and eigenmodes should be known and taken into account. Structure A portrays the structure in its undamaged condition, whereas structure B depicts the same structure with damage to the fourth, fifth, sixth, seventh, eighth and ninth elements of 20%, 30%, 40%, 60%, and 30%, respectively.

Table 4 Modal properties for structure A

		1st Eigenmode	2nd Eigenmode	3rd Eigenmode	4th Eigenmode
	$\lambda = \omega^2$ (sec^{-2})	**3008.56**	**48108.77**	**243219.45**	**765859.94**
Nodal values	**Eigenperiod *T* (sec)**	**0.1146**	**0.0286**	**0.0127**	**0.0072**
	1st DOF	0.505	-0.960	-1.321	-1.553
	2nd DOF	0.960	-1.553	-1.553	-0.960
	3rd DOF	1.322	-1.553	-0.505	0.960
	4th DOF	1.553	-0.959	0.960	1.553
	5th DOF	1.633	0.001	1.633	0.000
	6th DOF	1.553	0.960	0.960	-1.553
	7th DOF	1.321	1.553	-0.505	-0.960
	8th DOF	0.959	1.553	-1.553	0.960
	9th DOF	0.504	0.960	-1.321	1.553

Table 5 Modal properties for structure B

		1st Eigenmode	2nd Eigenmode	3rd Eigenmode	4th Eigenmode
	$\lambda = \omega^2$ (sec^{-2})	**1762.18**	**32163.04**	**174959.58**	**540463.95**
Nodal values	**Eigenperiod T (sec)**	**0.1497**	**0.0350**	**0.0150**	**0.0085**
	1st DOF	0.439	-0.854	-1.206	-1.438
	2nd DOF	0.849	-1.442	-1.552	-1.157
	3rd DOF	1.206	-1.585	-0.780	0.543
	4th DOF	1.476	-1.184	0.636	1.611
	5th DOF	1.627	-0.298	1.631	0.488
	6th DOF	1.629	0.787	1.269	-1.489
	7th DOF	1.441	1.576	-0.366	-1.164
	8th DOF	1.053	1.594	-1.582	1.035
	9th DOF	0.546	0.935	-1.245	1.468

Table 4 and Table 5 show the eigenproperties of the two structures. The numerical work in this section has been carried out in Excel using equations (3), (4), (5), (6), (9), and (10).

The Modal Assurance Criterion (MAC)

The first criterion used is the MAC criterion. This criterion indicates the correlation between the two sets of natural modes for structures A and B using equation (3).

MAC is a vector with four eigenmodes in our case, which has as many values as the number of eigenmodes. Considering all four eigenmodes in our case, we arrive to the following MAC values:

$$\mathrm{MAC(A, B)} = \mathrm{MAC(B, A)} = [0.9950 \quad 0.9853 \quad 0.9765 \quad 0.9609]$$

In order to obtain the final MAC scalar values, we multiply the individual values of the MAC vector using and get the following:

Table 6 MAC scalar values

No of known modes	**1**	**2**	**3**	**4**
MAC(A, B)	0.9950	0.9803	0.9573	0.9199

For our example, Table 6 displays the values for MAC for different values of the number of known eigenmodes. It can be noticed that with one eigenmode, the MAC results show a consistent correspondence between the two mode shape vectors, while with increasing number of known modes, the results a decrease in MAC values indicating a less consistent correspondence between the two mode shape vectors.

The Modified Total Modal Assurance Criterion (MTMAC)

The second criterion used is the MTMAC. The MTMAC criterion takes the eigenvalues of the different mode shapes of the structure in addition to the eigenvectors, which helps give us more correct results in case of a uniform damage in the structure. The criterion using equation (5) indicates the correlation between the two sets of natural modes for structures A and B.

MTMAC is a row vector with four eigenmodes in our case and considering all four eigenmodes, we arrive to the following MTMAC values:

$$\mathrm{MTMAC(A, B)} = \mathrm{MTMAC(B, A)} = [0.7889 \quad 0.8220 \quad 0.8395 \quad 0.8195]$$

And in order to acquire the final MTMAC values, we multiply the individual values of the MTMAC vector using equation (6) and get the following:

Table 7 MTMAC scalar values

No of known modes	**1**	**2**	**3**	**4**
MTMAC(A, B)	0.7889	0.6484	0.5444	0.4461

Table 7 shows the calculated values of MTMAC for various known eigenmodes. Similar to the MAC criterion, an index equal to zero means no correlation between the two sets of mode shapes and natural frequencies, whereas a value close to one shows a good correlation between the two sets.

The Co-ordinate Modal Assurance Criterion (CoMAC)

Another criterion, the Co-ordinate Modal Assurance criterion (COMAC), an extension of MAC is used in this example to determine the difference between the two modal vectors at the degree of freedom level. Using equation (4) the COMAC scalar is calculated, and the following values are obtained:

Table 8 COMAC scalar value

No of known modes	1	2	3	4
1st DOF	1.0000	0.9999	0.9997	0.9997
2nd DOF	1.0000	0.9995	0.9979	0.9905
3rd DOF	1.0000	0.9970	0.9804	0.9509
4th DOF	1.0000	0.9850	0.9623	0.9754
5th DOF	1.0000	0.9678	0.9837	0.9421
6th DOF	1.0000	0.9892	0.9760	0.9814
7th DOF	1.0000	0.9987	0.9934	0.9906
8th DOF	1.0000	0.9991	0.9993	0.9992
9th DOF	1.0000	0.9980	0.9983	0.9990
COMAC	**1.0000**	**0.9356**	**0.8955**	**0.8395**

Table 8 presents the COMAC scalar value for known eigenmodes for our example. When a COMAC value is low (less than 1), it suggests that the two sets of mode shapes for the chosen degree of freedom are inconsistent or dissimilar from one another, while a value closer to 1 or equal to 1 suggests that the two sets of mode shapes are concordant for that degree of freedom.

The Modal flexibility Assurance Criterion (MACFLEX)

Finally, the modal flexibility assurance criterion (MACFLEX) is applied to the two sets of mode shapes for structures A and B. MACFLEX compares the values of the flexibility matrix of a structure as damage affects the stiffness matrix of a structure and reduces its stiffness. Using equation (9) the dynamically measured flexibility calculated from the identified modal parameters is calculated. The MACFLEX scalar value is finally calculated using equation (10).

Table 9 shows the MACFLEX scalar value for 1, 2, 3 and 4 known eigenmodes, for our example. A score near to one, as with the other aforementioned criteria, signifies almost little change in flexibility and as a result, a perfect correlation between the two structures, A and B. In contrast, a value close to zero denotes no correlation between the two structures. With an increasing number of known modes, for our example, we can see that the MACFLEX value rises and becomes closer to one.

Table 9 MACFLEX scalar value

No of known modes	**1**	**2**	**3**	**4**
1st DOF	0.9950	0.9929	0.9932	0.9933
2nd DOF	0.9950	0.9936	0.9939	0.9939
3rd DOF	0.9950	0.9947	0.9948	0.9948
4th DOF	0.9950	0.9957	0.9956	0.9956
5th DOF	0.9950	0.9963	0.9962	0.9962
6th DOF	0.9950	0.9965	0.9965	0.9965
7th DOF	0.9950	0.9967	0.9967	0.9967
8th DOF	0.9950	0.9972	0.9973	0.9973
9th DOF	0.9950	0.9978	0.9980	0.9979
MACFLEX	**0.9557**	**0.9621**	**0.9628**	**0.9629**

4. Case Study

The objective of this thesis is to explore a method for accurately detecting both the position and extent of damage in structures. The focus will be on utilizing specific dynamic characteristics data of the structure that can be easily measured. The data of the structure is acquired by developing software that performs the dynamic analysis of structures using the FEM. The investigation will specifically target a 10-bar plane truss as the structure for damage identification.

To address the problem of structural damage identification, an optimization approach is employed, utilizing the GA (Genetic Algorithm) and PSO (Particle Swarm Optimization) algorithms. These algorithms are introduced in Chapter 2.4, and their effectiveness is examined in Chapter 3.1 of the literature review.

Furthermore, this thesis delves into the assessment of various modal correlation criteria in relation to structural damage. Two modal correlation criteria, namely MACFLEX and MTMAC, are employed as objective functions within the optimization process to determine the location and extent of the structural damage. The performance of each objective function is evaluated across multiple damage scenarios involving a 10-bare plane truss structure.

5. Methodology

5.1. Problem definition

Three key elements in the structural damage diagnosis process are the establishment of an objective function, choosing the update parameters, and using reliable optimization techniques. They demand profound physical understanding and often use trial-and-error techniques. For our case, the problem of identifying damage is viewed as an unconstrained optimization problem, where the extent of damage for each individual element within the structure is represented by the design variables. Consequently, the number of design variables corresponds to the total number of elements present in the structure.

In the current thesis, the modal parameters computed from both the experimental and numerical models serve as the objective function for damage detection. The dynamic characteristics data collected by certain experiments make up the experimental model. However, in the current thesis, this information is not available, necessitating an alternative approach to obtain the experimental values of natural frequencies and the mode shapes of the structure. Here is the procedure used:

- Firstly, it is presumed to have pre-existing knowledge of the damage index values. Damage extent is quantified using a scalar variable or index denoted by d, which ranges from 0, indicating no damage, to 1, representing 100% damage. However, for numerical stability reasons, the maximal damage value is set at 0.999, barely below 1. This precaution is taken to prevent the structure from turning into an unanalyzable mechanism, which could result in numerical instability.
- Using equation (2), the modal parameters of the structure are computed. These modal parameters correspond to the exact actual values and are difficult to attain through experimental measurements. It is commonly presumed that the vibration frequencies have been determined accurately during modal testing, with experimental errors occurring predominantly in the determination of mode shape amplitudes. To better align calculated modal parameters with actual experimental values, noise is added to the calculated parameters or each mode shape using equation [38]:

$$V_{num} = V_{num} * (1 + 0.01 * NoiseRatio * \xi) \quad (23)$$

By integrating these stages, the objective function for damage detection incorporates both numerical and experimental modal parameters, thereby providing a holistic approach to the thesis.

The damage index values, which are not preset, are now used in the numerical model to represent the design variables of the optimization problem. The numerical stiffness matrix and modal parameters are generated for each iteration of the optimization procedure, which starts with the damage index being given random values. The reduction of the discrepancy between the modal parameter values derived from the numerical and experimental models is the main goal of the minimization procedure in the objective function.

In this paper, three objective functions were taken into account corresponding to the MTMAC and MACFLEX correlation requirements between the actual damage and damage calculated by FEM and are presented as follows:

$$y_1 = 100 \cdot (1 - MACFLEX) \tag{24}$$

$$y_2 = 1 - MTMAC \tag{25}$$

$$y_3 = \sqrt{y_1^2 + y_2^2} \tag{26}$$

The third objective function (y_3) is employed to combine the benefits of the first two objective functions.

The minimum target value for all the objective functions are set to zero. In practical situations, the dynamic properties, such as eigenvalues and eigenmodes of the actual damaged structure need to be measured experimentally in order to be determined. In our case, however, these properties are calculated numerically utilizing a "real damage" finite element model.

As mentioned earlier, the problem of identifying damage is viewed as an unconstrained optimization problem, and this problem is solved using two optimization algorithms, GA and PSO.

5.2. Finite Element Analysis

Sirois and Grilli define numerical modelling as "a mathematical representation of a physical behavior, based on relevant hypotheses and simplifying assumptions [39]." Engineers throughout the world are utilizing numerical modelling and computer simulations to solve engineering problems. This approach is applicable to a wide range of problems, spanning from simple 2D problems to intricate 3D problems and abstract models. While it is theoretically possible to solve these problems manually or through outdated and less advanced methods, such approaches would take a significantly larger amount of time compared to using numerical modeling and computer simulations.

There are numerous varieties of numerical modeling. This thesis employs the finite element method (FEM), one of the most widely used methodologies, for numerical modeling. When applied to a simulation, this technique is known as FEA – Finite Element Analysis. FEA demonstrates how the elements in a structure react and perform when subjected to stress and loads [40].

In order to study and analyze the behavior of the truss structure used in this study, Matlab software is utilized. A program has been developed in the software and it is written in Matlab programming language. To perform dynamic analysis, the program needs an input file inserted by the user. The file is a text file (in text format) that includes structural data such as the length of the structure, the assigned load, section area, and the modulus of elasticity. The program draws and displays the model in a simple figure. The analysis's findings are then recorded in a text-based output file. Modal correlation criteria, specifically the MTMAC and MACFLEX, is also integrated into the program in order to compare the sets of values for the experimental and numerical structures. Finally, the two optimization algorithms, GA and PSO, adopted in this study to solve the optimization problems are integrated into the program.

6. Numerical Example – 10-bar plane truss

6.1. Model Input

A 10-bar plane truss model is analyzed in this thesis to illustrate the performance of the proposed methodology and the different criteria. The 2D truss model is presented in Figure 2 10-bar plane truss model. The Young's modulus of the truss is $2.1*10^8$ kN/m^2, length L is 6m, the assigned load P is 100 kN and the material mass m is 0.5 kg. The structural members are divided into 10 groups and their section area is shown in Table 10.

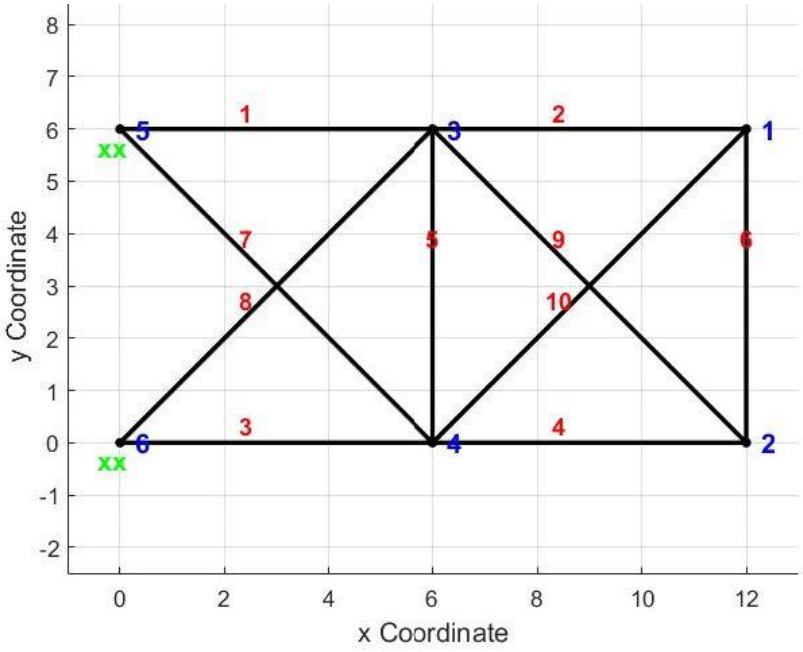

Figure 2 10-bar plane truss model

Table 10 Section Area for each member m^2

Member ID	Section Area (m^2)
1	0.00200
2	0.00200
3	0.00200
4	0.00200
5	0.00200
6	0.00200
7	0.00200
8	0.00200
9	0.00200
10	0.00200

Table 11 General optimization parameter for 10-bar plane truss

Design variable	**d (damage index)**
Number of design variables	10
Lower bound (lb)	0
Upper bound (ub)	0.999

Table 12 GA parameter values for 10-bar plane truss

Matlab command	*ga*
Number of variables (D)	10
Number of runs	5
Generations	100
Population size (NP)	200

Table 13 PSO parameter values for 10-bar plane truss

Matlab command	*particleswarm*
Number of variables (D)	10
Number of runs	5
Swarm size	200
MaxIterations	100

The structure is exposed to three damage scenarios in order to study the performance of the suggested methodology. These scenarios are as follows:

1. 40% damage at element 1
2. 20%, 40% and 60% at element 4, 8 and 9 respectively
3. A uniform damage of 20% at all elements

Objective functions:

$$y_1 = 100 \cdot (1 - MACFLEX)$$

$$y_2 = 1 - MTMAC$$

$$y_3 = \sqrt{y_1^2 + y_2^2}$$

Noise:

$$V_{num} = V_{num} * (1 + 0.01 * NoiseRatio * \xi)$$

6.2. Results

All damage scenarios have been optimized using the same optimizing algorithms, GA and PSO. Both optimization algorithms run 5 times and the best result is obtained by the run with the minimum objective function value. To maintain consistency for all the tests, for both GA and PSO, the maximum number of iterations is set to 200.

Two different modal correlation criteria (MACFLEX and MTMAC) are used to formulate the three objective functions (24), (25) and (26). The range of known eigenmodes for each criterion ranges from 1 to 4.

Both tables and bar charts are used to display the findings. The real damage, which represents the target damage, is always shown in red color in bar charts, whereas the colored bars represent calculated damage based on the optimization algorithms.

6.2.1. Damage scenario 1

The first damage scenario is going to be 40% at element 1. Multiple tests with the three objective functions are carried out within this damage scenario, namely equations (24), (25) and (26). The first set of tests is carried out without the application of noise to each mode shape. The second set is carried out with the application of 5% and 10% noise to the mode shapes of the structure. Tests were carried out for 1, 2, 3 and 4 known eigenmodes.

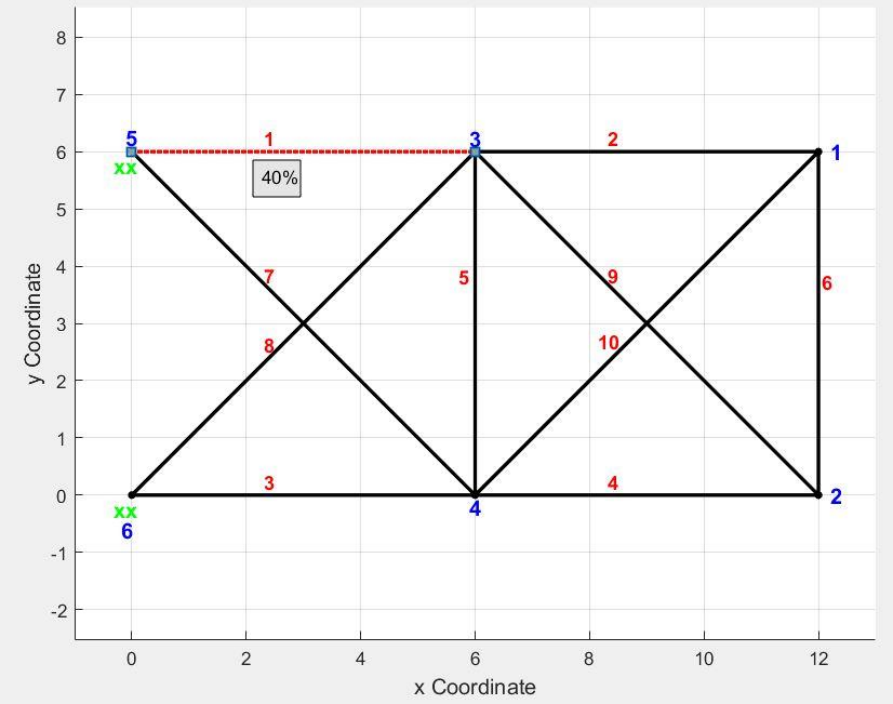

Figure 3 10-bar plane truss with single element damage

i) Without the application of noise.

a) GA Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 14 Optimization results in percentage with GA for single element damage using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	40.0	0	0	0	0	0	0	0	0	0
1 mode	76.04	59.64	57.87	58.88	42.78	84.54	56.68	62.92	58.41	59.63
2 modes	65.93	43.18	43.30	43.06	43.23	43.39	43.21	43.18	43.26	43.17
3 modes	64.52	40.94	40.93	40.95	40.87	40.58	40.91	40.96	40.98	40.90
4 modes	65.82	43.08	43.02	43.06	42.9	42.9	43.0	43.05	43.0	42.9

Mode	Min. objective function value
1	1.15E-04
2	2.12E-05
3	1.59E-05
4	4.58E-06

b) PSO Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 15 Optimization results in percentage with PSO for single element damage using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	40.0	0	0	0	0	0	0	0	0	0
1 mode	40.0	0	0	0	0	0	0	0	0	0
2 modes	40.0	0	0	0	0	0	0	0	0	0
3 modes	40.0	0	0	0	0	0	0	0	0	0
4 modes	40.0	0	0	0	0	0	0	0	0	0

Mode	Min. objective function value
1	2.26E-09
2	1.44E-13
3	1.33E-12
4	2.73E-10

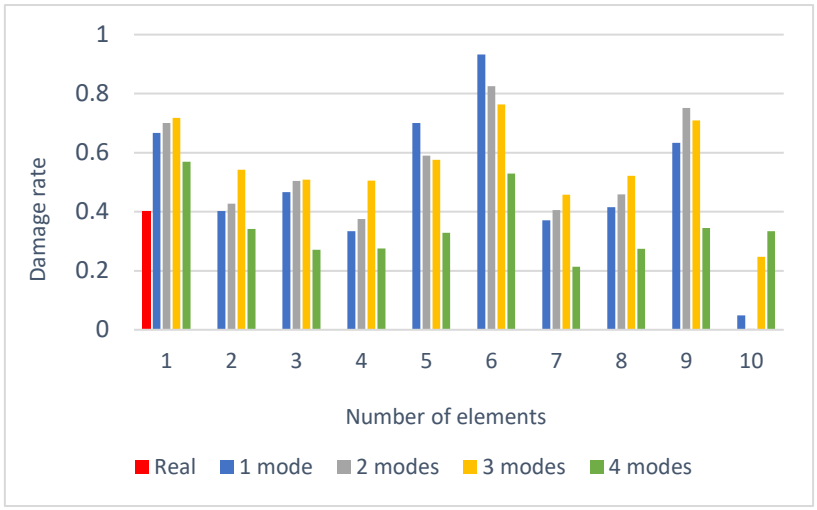

Figure 4 Damage results - Real vs Calculated (GA) using objective function y_1.

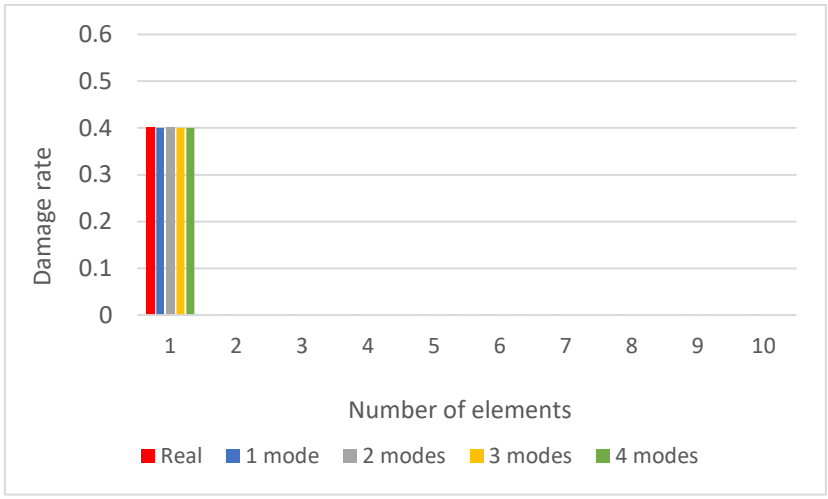

Figure 5 Damage results - Real vs Calculated (PSO) using objective function y_1.

Figure 4 and Figure 5 shows how the first objective function (y_1) performs for the first damage scenario using GA and PSO algorithms. We can see that the PSO algorithm shows a better performance, since it manages to identify the damage 100% in all modes known. This is also seen in Table 15, as the algorithm scored 100% identical damage percentage to that of the real damage. The GA algorithm, however, shows poor performance in identifying the extent and location of the damage. Although it gives an indication of damage to the existence in the structure, it is not a valid damage rate for the specific element.

c) GA Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 16 Optimization results in percentage with GA for single element damage using objective function y_2.

	1	2	3	4	5	6	7	8	9	10
Real	40.0	0	0	0	0	0	0	0	0	0
1 mode	29.3	33.4	2.85	15.3	72.4	85.3	5.2	15.2	1.3	21.8
2 modes	32.5	8.45	2.5	1.46	3.21	0.9	4.2	3.2	19.2	21.8
3 modes	35.8	1	1.9	0	9.15	17.8	0.4	1.1	28.5	1.3
4 modes	39.6	0.2	0.6	0	0.3	1.7	0	0	0	0.3

Mode	**Min. objective function value**
1	0.000587
2	0.0012904
3	0.00666
4	0.001755

d) PSO Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 17 Optimization results in percentage with PSO for single element damage using objective function y_2.

	1	2	3	4	5	6	7	8	9	10
Real	40.0	0	0	0	0	0	0	0	0	0
1 mode	39.5	1	0	0.7	9.6	52.4	0	0.1	2.7	0.8
2 modes	39.9	0	0	0	0	0	0	0	0	0.1
3 modes	40.0	0	0	0	0	0	0	0	0	0
4 modes	40.0	0	0	0	0	0	0	0	0	0

Mode	**Min. objective function value**
1	2.07E-06
2	3.00E-08
3	2.18E-09
4	5.89E-10

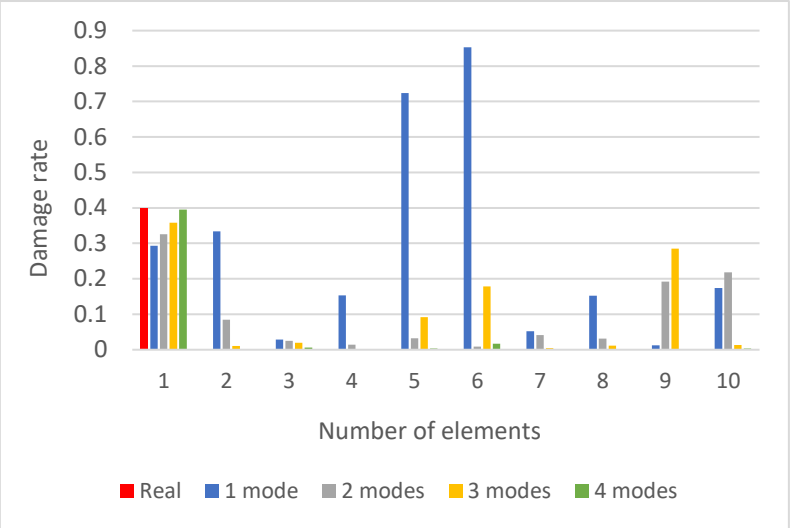

Figure 6 Damage results - Real vs Calculated (GA) using objective function y_2.

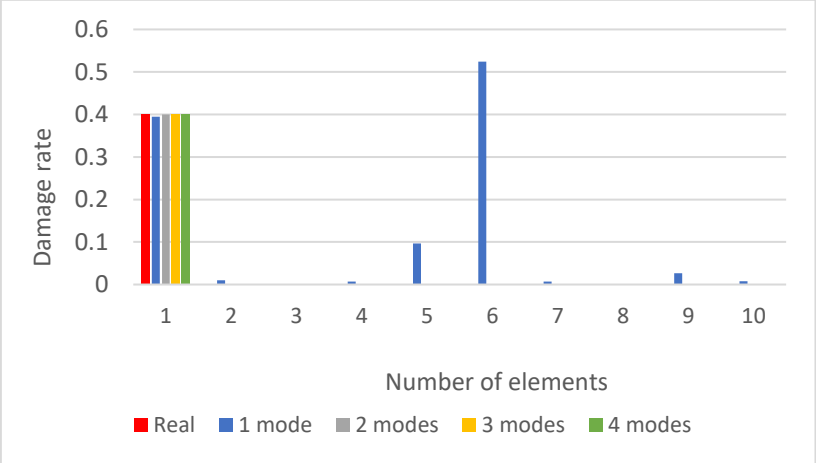

Figure 7 Damage results - Real vs Calculated (PSO) using objective function y_2.

Figure 6 and Figure 7 shows the performance of the second objective function (y_2) for damage scenario 1, using both the GA and PSO algorithms. Again, we see that the PSO algorithm performs better than the GA in identifying the extent and location of the damage. We can also see that the algorithm exhibits damage in element 6 with 1 known mode, but with more known modes, the algorithm is able to provide with the exact damage extent and location as can be seen in Table 17. The GA algorithm show a very good performance with this second objective function, especially when 4 eigenmodes are known, since it manages to identify the damage almost 100%.

e) GA Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 18 Optimization results in percentage with GA for single element damage using objective function y_3.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	40.0	0	0	0	0	0	0	0	0	0
1 mode	41.4	0.9	0.7	1.7	2.6	6.6	1.36	2.8	1.5	2.7
2 modes	55.9	36.8	33.2	22.3	27.2	26.7	34.6	27.9	30.7	30.3
3 modes	39.6	0.9	0.8	2.4	3.1	2.1	1.4	1	2.4	1.9
4 modes	39.7	1.2	0.8	1.5	0.4	5.6	0.9	1	1.2	1.9

Mode	**Min. objective function value**
1	0.008534
2	0.28665
3	0.014822
4	0.017381

f) PSO Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 19 Optimization results in percentage with PSO for single element damage using objective function y_3.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	40.0	0	0	0	0	0	0	0	0	0
1 mode	40.0	0	0	0	0.2	1.1	0	0	0	0
2 modes	39.9	0	0	0	0	0.2	0	0	0	0
3 modes	40.0	0	0	0	0	0	0	0	0	0
4 modes	40.0	0	0	0	0	0	0	0	0	0

Mode	**Min. objective function value**
1	1.69E-07
2	6.34E-07
3	4.69E-09
4	1.71E-10

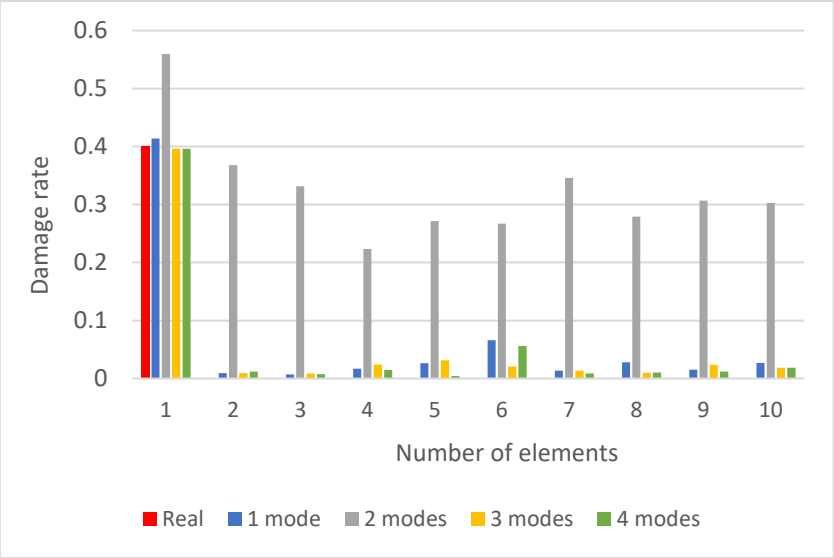

Figure 8 Damage results - Real vs Calculated (GA) using objective function y_3.

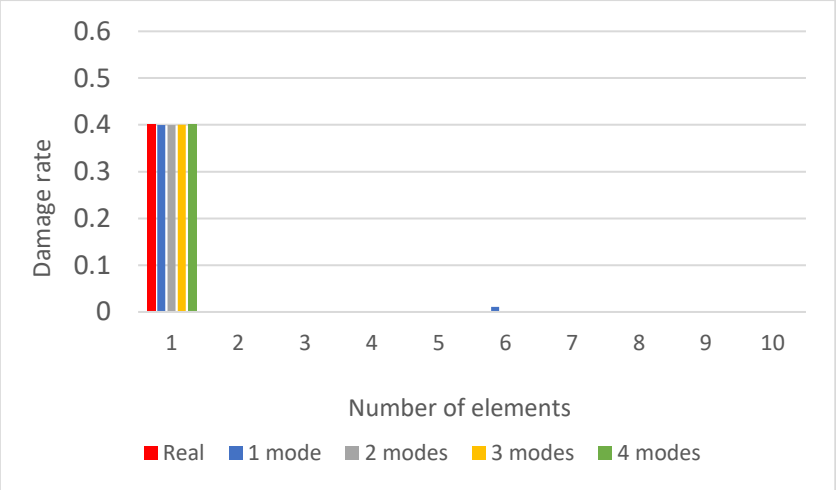

Figure 9 Damage results - Real vs Calculated (PSO) using objective function y_3.

Figure 8 and Figure 9 shows the performance of the third objective function (y_3) for the first damage scenario using GA and PSO algorithms, respectively. Both algorithms show a very good performance in identifying the damage. Again, the same trend shows an excellent performance for the PSO when compared to that of the GA. The GA algorithm with the third objective function gives a clearer indication about the position and the rate of the damage, especially with 1, 3 and 4 modes known, when compared to the results with the first and second objective functions.

ii) With applied noise of 5%

a) GA Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 20 Optimization results in percentage with GA for single element damage using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	40.0	0	0	0	0	0	0	0	0	0
1 mode	66.7	40.3	46.6	33.4	70.0	93.3	37.1	41.5	63.3	4.8
2 modes	70.1	42.7	50.5	37.5	59.0	82.5	40.5	45.9	75.2	0.3
3 modes	71.8	54.2	50.9	50.5	57.6	76.4	45.8	52.2	70.9	24.7
4 modes	56.9	34.2	27.2	27.6	32.9	52.9	21.4	27.5	34.5	33.4

Mode	**Min. objective function value**
1	0.000153
2	3.73E-01
3	5.68E-01
4	0.57772

b) PSO Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 21 Optimization results in percentage with PSO for single element damage with 5% added noise using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	40	0	0	0	0	0	0	0	0	0
1 mode	62.6	36.7	34.8	11.7	3.2	97.8	19.1	42.1	49.8	0
2 modes	64.6	32.9	41.1	28.0	50.5	78.2	30.0	35.9	69.5	0
3 modes	60.6	35.8	31.3	31.7	40.1	66.8	24.4	32.9	58.9	0
4 modes	45.4	16.7	7.6	7.9	14.4	41.8	0.4	8.2	19.3	15.3

Mode	**Min. objective function value**
1	3.38E-07
2	0.35951
3	0.56993
4	0.57669

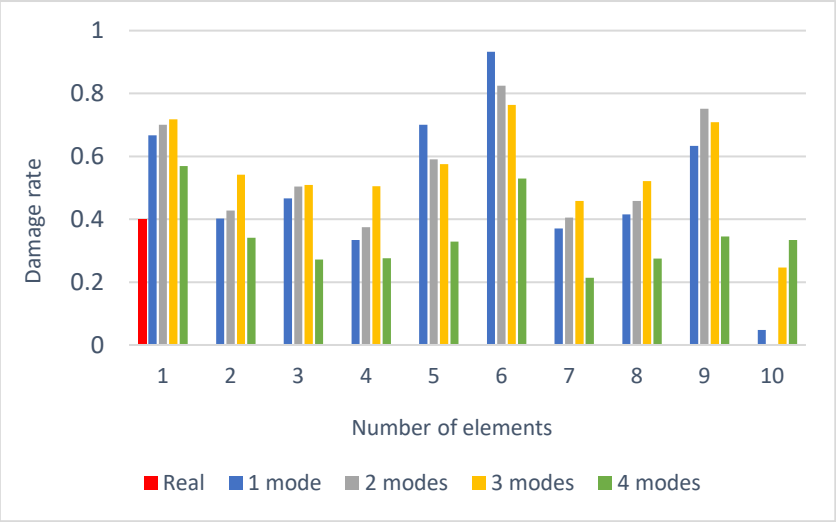

Figure 10 Damage results - Real vs Calculated (GA) using objective function y_1 with 5% noise.

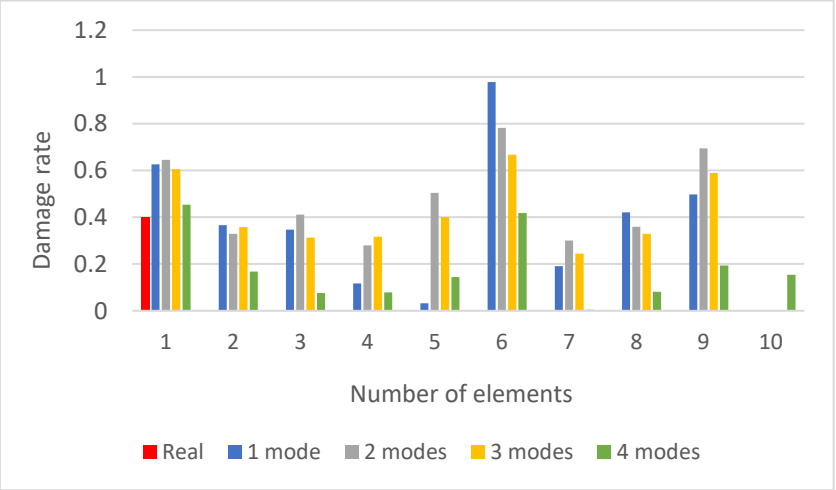

Figure 11 Damage results - Real vs Calculated (PSO) using objective function y_1 with 5% noise.

Figure 10 and Figure 11 shows the performance of the first objective function (y_1) for the first damage scenario, using both GA and PSO algorithms, when noise is applied to the mode shapes of the structure. The value of the noise ratio is set at 5%. Both algorithms appear not to exhibit a very good performance, even in the cases where 4 eigenmodes are known.

c) GA Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 22 Optimization results in percentage with GA for single element damage with 5% added noise using objective function y_2.

	1	2	3	4	5	6	7	8	9	10
Real	40.0	0	0	0	0	0	0	0	0	0
1 mode	33.3	14.2	0.9	17.2	79.1	99.9	9.6	5.1	11.8	1.1
2 modes	26.6	18.6	0.9	0.1	14.8	0.3	1.9	0	34.2	41.8
3 modes	33.6	7.5	3.8	0.1	4.9	5.0	0	0.4	31.9	5.6
4 modes	38.9	0.4	1.5	0.3	1.1	6.0	0.7	0	0.3	0.2

Mode	**Min. objective function value**
1	0.0005513
2	0.0042633
3	0.012018
4	0.009182

d) PSO Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 23 Optimization results in percentage with PSO for single element damage with 5% added noise using objective function y_2.

	1	2	3	4	5	6	7	8	9	10
Real	40.0	0	0	0	0	0	0	0	0	0
1 mode	36.9	7.9	0.4	2.2	58.0	99.9	0.1	0	20.4	0
2 modes	37.0	6.7	0	0	0	1.3	0.1	0.3	8.5	18.7
3 modes	40	0	0	0	0	0	0	0	0	0
4 modes	40	0	0	0	3.2	0	0	0	0	0

Mode	**Min. objective function value**
1	0.000217
2	0.001324
3	0.00295
4	0.003708

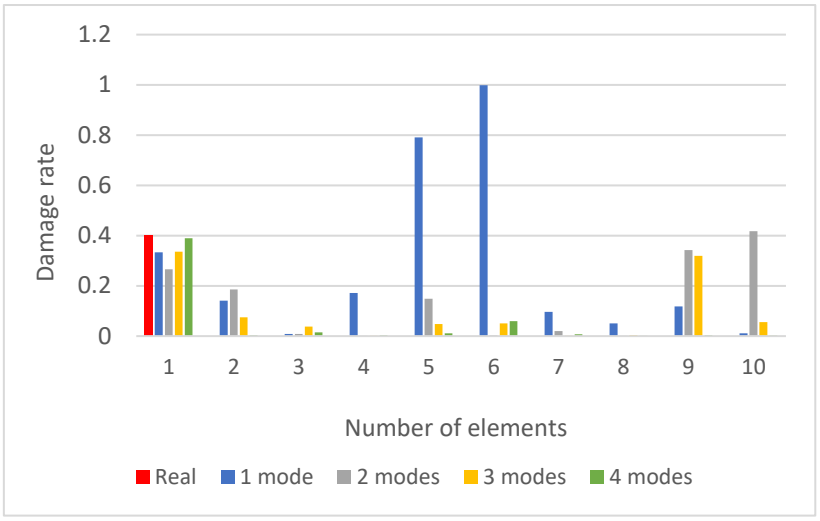

Figure 12 Damage results - Real vs Calculated (GA) using objective function y_2 with 5% noise.

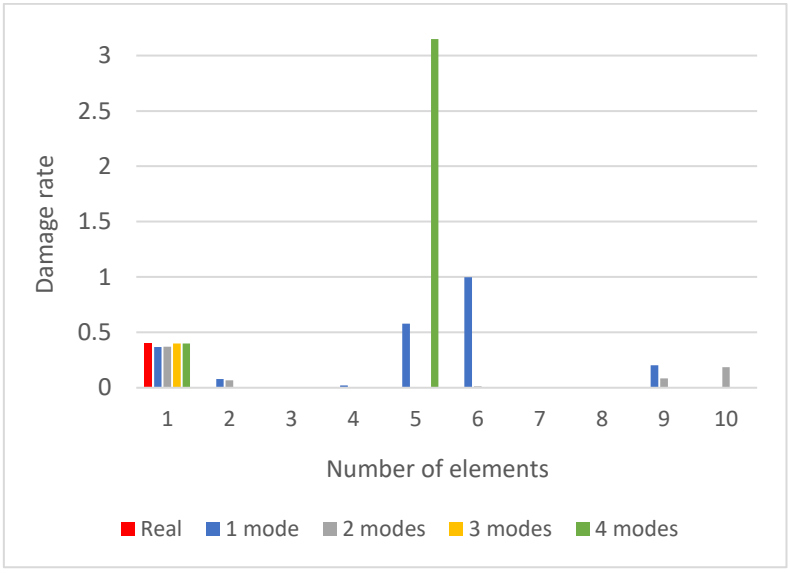

Figure 13 Damage results - Real vs Calculated (PSO) using objective function y_2 with 5% noise.

Figure 12 and Figure 13 show the performance of the second objective function (y_2) for the first damage scenario, when noise of 5% is applied to the mode shapes of the structure. It can be seen that even though the application of noise in the calculated data influences its values, both algorithms performed good in identifying the extent and location of the damage. Again here, we see that the PSO algorithm outperforms the GA in giving better results, with more eigenmodes known.

e) GA Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 24 Optimization results in percentage with GA for single element damage with 10% added noise using objective function y_3.

	1	2	3	4	5	6	7	8	9	10
Real	40.0	0	0	0	0	0	0	0	0	0
1 mode	53.3	20.4	16.9	0.6	4.9	99.2	0.3	27.4	36.0	0.5
2 modes	62.8	30.9	38.3	23.7	48.8	77.9	27.6	33.1	66.8	0.4
3 modes	54.4	25.5	20.6	22.7	30.9	61.5	12.7	22.3	48.8	0.4
4 modes	46.9	17.8	10.3	10.8	21.5	41.4	3.6	10.1	19.3	15.7

Mode	Min. objective function value
1	0.091732
2	0.49568
3	0.66493
4	0.64594

f) PSO Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 25 Optimization results in percentage with PSO for single element damage with 5% added noise using objective function y_3.

	1	2	3	4	5	6	7	8	9	10
Real	40.0	0	0	0	0	0	0	0	0	0
1 mode	46.9	8.3	0	0	27.4	99.9	0	7.1	27.2	0
2 modes	56.3	16.9	26.3	0	39.9	74.8	14.4	0	56.9	0
3 modes	50.9	20.8	14.7	17.5	0	0	0	0	42.6	0
4 modes	42.7	12.6	4.2	4.0	10.5	37.3	0	3.8	12.9	8.9

Mode	Min. objective function value
1	7.01E-02
2	4.85E-01
3	6.58E-01
4	6.13E-01

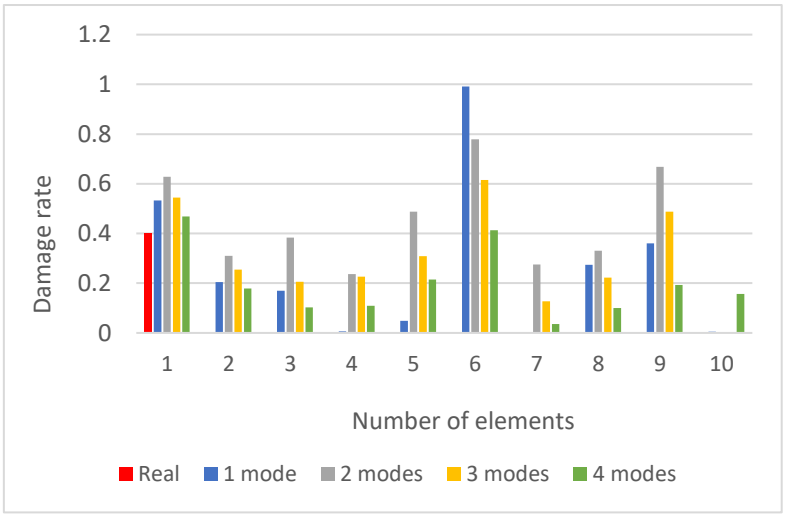

Figure 14 Damage results - Real vs Calculated (GA) using objective function y_3 with 5% noise.

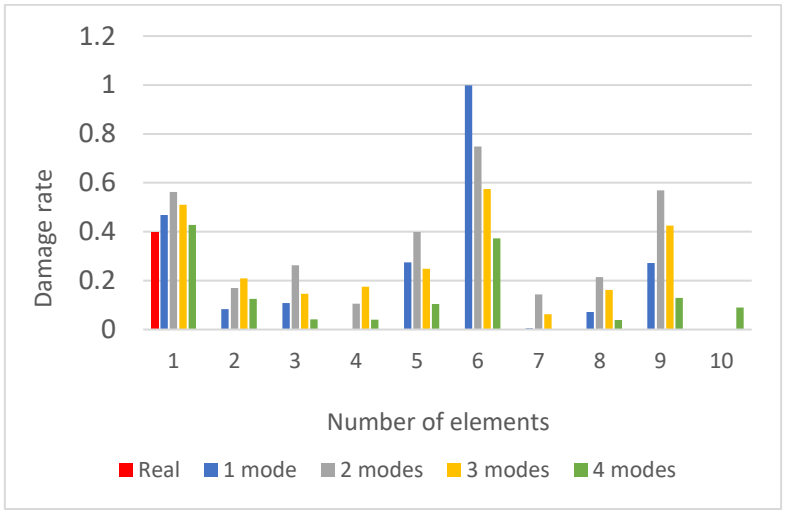

Figure 15 Damage results - Real vs Calculated (PSO) using objective function y_3 with 5% noise.

Figure 14 and Figure 15 show the performance of the third objective function (y_3) for the first damage scenario when a noise of 5% is applied to the mode shape of the structure. Similar to the first objective function, we can see that both algorithms does not exhibit a good performance in identifying the real damage. The PSO algorithm, however, shows reasonable results with 4 known eigenmodes.

iii) With applied noise of 10%

a) GA Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 26 Optimization results in percentage with GA for single element damage with 10% added noise using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	40.0	0	0	0	0	0	0	0	0	0
1 mode	76.0	63.5	66.6	25.6	85.6	99.9	57.4	48.5	75.4	0
2 modes	77.7	54.2	63.7	50.2	71.4	89.9	49.5	60.4	89.6	0.1
3 modes	73.1	58.1	52.5	54.9	63.7	81.8	42.5	54.1	80.1	0.4
4 modes	49.9	30.9	12.5	16.5	24.0	52.7	0	14.8	34.2	32.7

Mode	**Min. objective function value**
1	0.0056
2	9.88E-01
3	2.12E+00
4	2.4065

b) PSO Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 27 Optimization results in percentage with PSO for single element damage with 10% added noise using objective function y_1.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	40	0	0	0	0	0	0	0	0	0
1 mode	76.9	59.1	56.2	14.8	21.7	99.9	32.9	68.8	71.0	0
2 modes	79.4	56.5	65.8	55.0	74.7	90.9	51.9	63.4	89.8	0.2
3 modes	75.2	61.6	56.8	58.6	66.9	83.8	46.9	57.5	81.8	0.3
4 modes	48.4	28.0	11.5	15.2	20.8	50.3	0	11.4	30.6	28.9

Mode	**Min. objective function value**
1	0.012292
2	0.95878
3	2.0988
4	2.4186

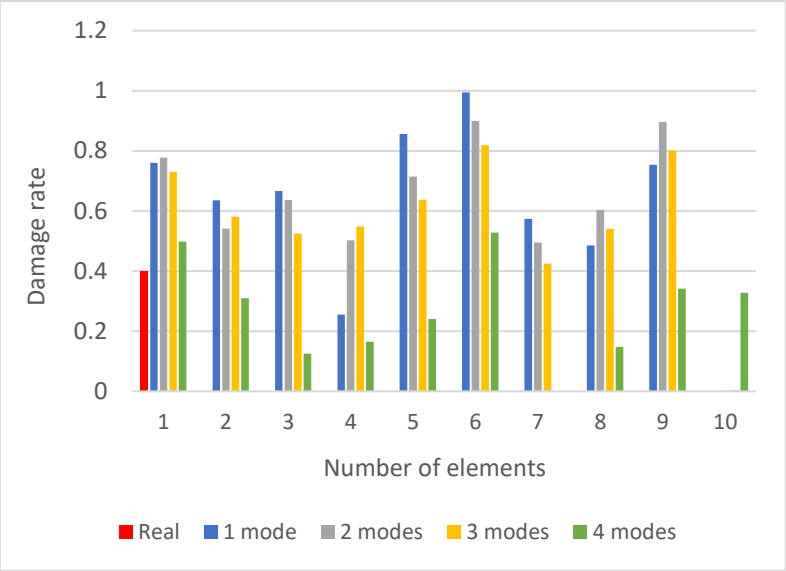

Figure 16 Damage results - Real vs Calculated (GA) using objective function y_1 with 10% noise.

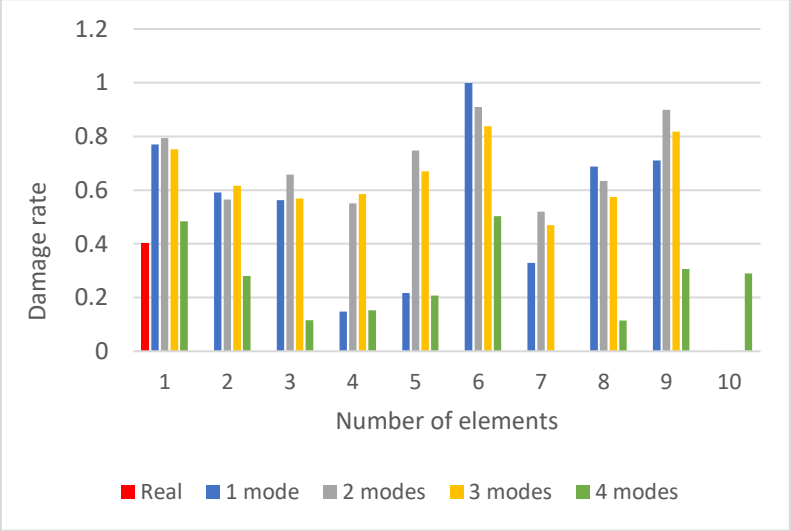

Figure 17 Damage results - Real vs Calculated (PSO) using objective function y_1 with 10% noise.

Figure 16 and Figure 17 show the performance of the first objective function (y_1) for the first damage scenario when a noise of 10% is applied to the mode shapes of the structure. With a high noise ratio, we see that both algorithms are not exhibiting a good performance, as they both indicate an existence of damage with various damage rates in all elements of the structure. The real damage is 40% at element 1, while both figures show a very high damage rate in elements 6 and 9, with 1, 2 and 3 known eigenmodes.

c) GA Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 28 Optimization results in percentage with GA for single element damage with 10% added noise using objective function y2.

	1	2	3	4	5	6	7	8	9	10
Real	40.0	0	0	0	0	0	0	0	0	0
1 mode	32.5	18.9	2.4	16.2	84.8	99.9	0.2	8.1	21.0	0.5
2 modes	1.2	15.0	17.2	19.1	13.6	9.6	5.6	32.1	9.6	47.4
3 modes	34.7	10.8	0.8	0.7	0.3	5.3	0.2	0.2	22.9	17.9
4 modes	39.7	0.6	0	0	0.8	6.6	0.2	0.6	0.7	0.8

Mode	**Min. objective function value**
1	0.001473
2	0.026197
3	0.021063
4	0.02028

d) PSO Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 29 Optimization results in percentage with PSO for single element damage with 10% added noise using objective function y_2.

	1	2	3	4	5	6	7	8	9	10
Real	40.0	0	0	0	0	0	0	0	0	0
1 mode	34.1	6.2	0	0	69.9	99.9	0	0	36.7	0
2 modes	34.6	12.3	0	0	0	1.9	0.7	0	20.2	22.9
3 modes	40	0	0	0	0	0	0	0	0	0
4 modes	40	0	0	0	0	0	0	0	0	0

Mode	**Min. objective function value**
1	0.000932
2	0.005346
3	0.011941
4	0.015084

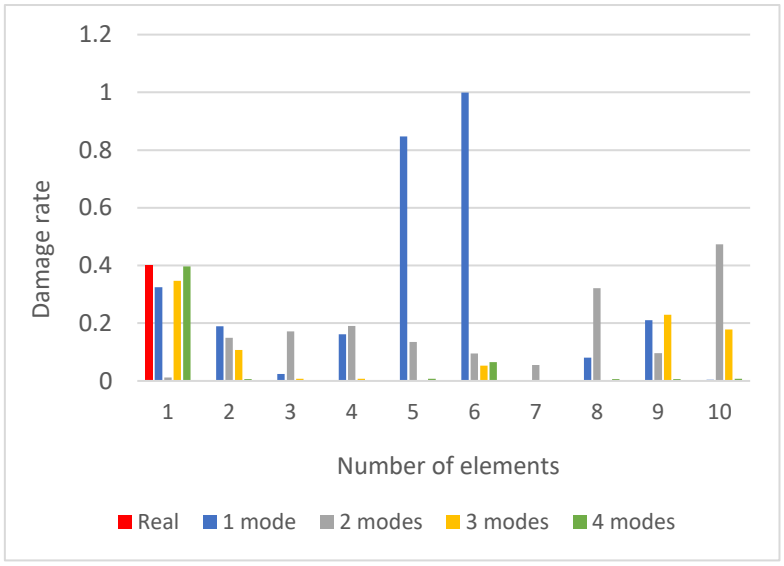

Figure 18 Damage results - Real vs Calculated (GA) using objective function y_2 with 10% noise.

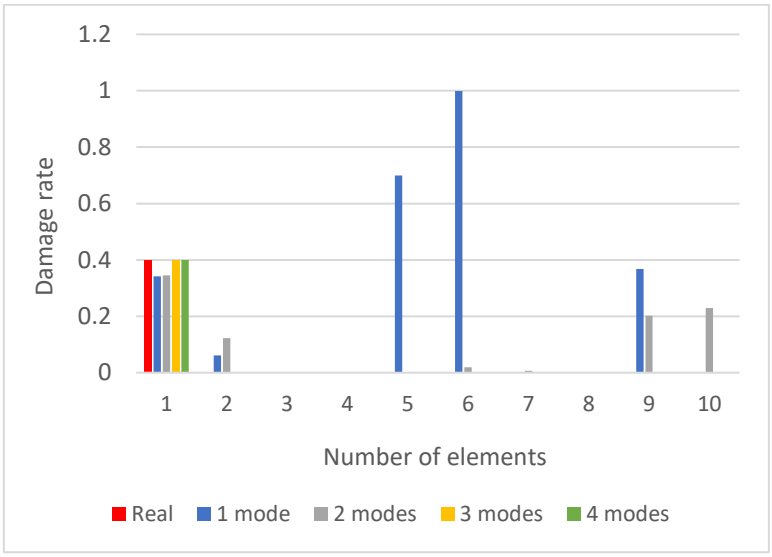

Figure 19 Damage results - Real vs Calculated (PSO) using objective function y_2 with 10% noise.

Figure 18 and Figure 19 show the performance of the second objective function (y_2) for the first damage scenario when a noise ratio of 10% is added to the mode shapes of the structure. There has been a pattern of an excellent performance of identifying the extent and location of the real damage with the second objective function, and as we can see here, even with applying a high noise rate of 10%, the GA algorithm exhibit a good performance, with 4 known eigenmodes and the PSO shows an exceptional performance in the cases where 3 and 4 eigenmodes are known.

e) GA Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 30 Optimization results in percentage with GA for single element damage with 10% added noise using objective function y_3.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	40.0	0	0	0	0	0	0	0	0	0
1 mode	64.4	34.9	35.9	0	30.1	99.9	9.3	43.6	57.6	0
2 modes	73.2	45.2	58.3	39.7	68.1	89.2	38.5	53.3	85.6	2.1
3 modes	70.5	53.9	47.8	51.4	60.4	80.2	36.6	49.3	77.9	0.3
4 modes	53.4	33.8	18.9	22.9	32.3	56.9	6.1	20.6	40.6	39.1

Mode	**Min. objective function value**
1	0.20365
2	1.2055
3	2.2213
4	2.4626

f) PSO Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 31 Optimization results in percentage with PSO for single element damage with 10% added noise using objective function y_3.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	40.0	0	0	0	0	0	0	0	0	0
1 mode	62.8	32.2	34.5	0	36.1	99.9	9.3	38.2	56.1	0
2 modes	77.3	52.9	62.9	48.9	71.2	90.0	47.7	59.2	89.2	0
3 modes	71.1	55.5	48.5	51.9	61.1	80.7	37.0	50.6	78.1	0
4 modes	46.6	26.6	7.9	11.3	18.1	50.3	0	9.9	30.6	29.3

Mode	**Min. objective function value**
1	2.03E-01
2	1.12E+00
3	2.22E+00
4	2.46E+00

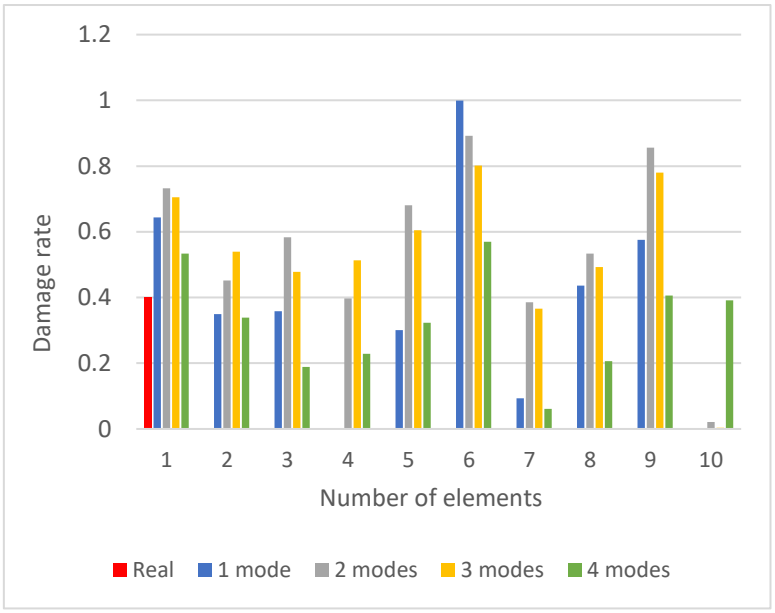

Figure 20 Damage results - Real vs Calculated (GA) using objective function y3 with 10% noise.

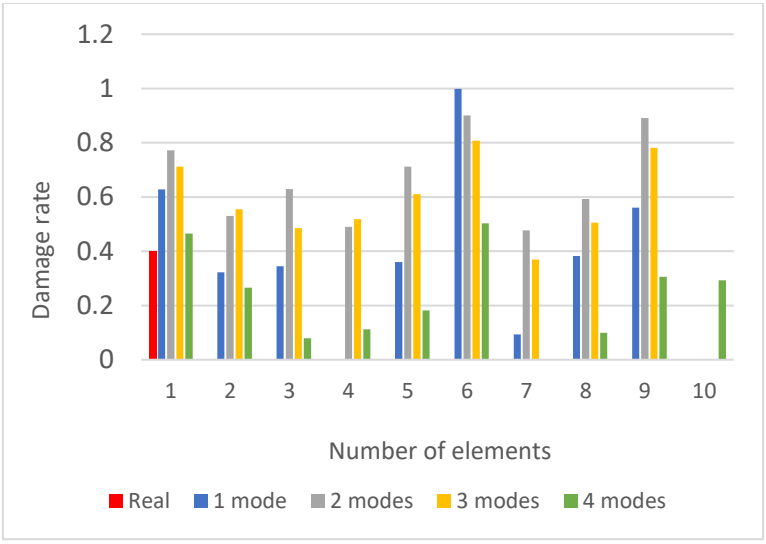

Figure 21 Damage results - Real vs Calculated (PSO) using objective function y_3 with 10% noise.

Figure 20 and Figure 21 show the final results for the first damage scenario, with the third objective function (y_3) and an applied noise of 10%. With this added noise ratio, both algorithms show a poor performance in identifying the exact extent and location of the real damage, even when 4 eigenmodes are known.

6.2.2. Damage scenario 2

The second damage scenario is going to be 20%, 40% and 60% at element 4, 8 and 9 respectively. Similarly, to the first damage scenario, multiple tests with the three objective functions are carried out using equations (24), (25) and (26) within this damage scenario. The first set of tests are analyzed without the application of noise to each mode shape. The second set with the application of 5% and 10% noise. Additionally, tests were carried out for 1, 2, 3 and 4 known eigenmodes.

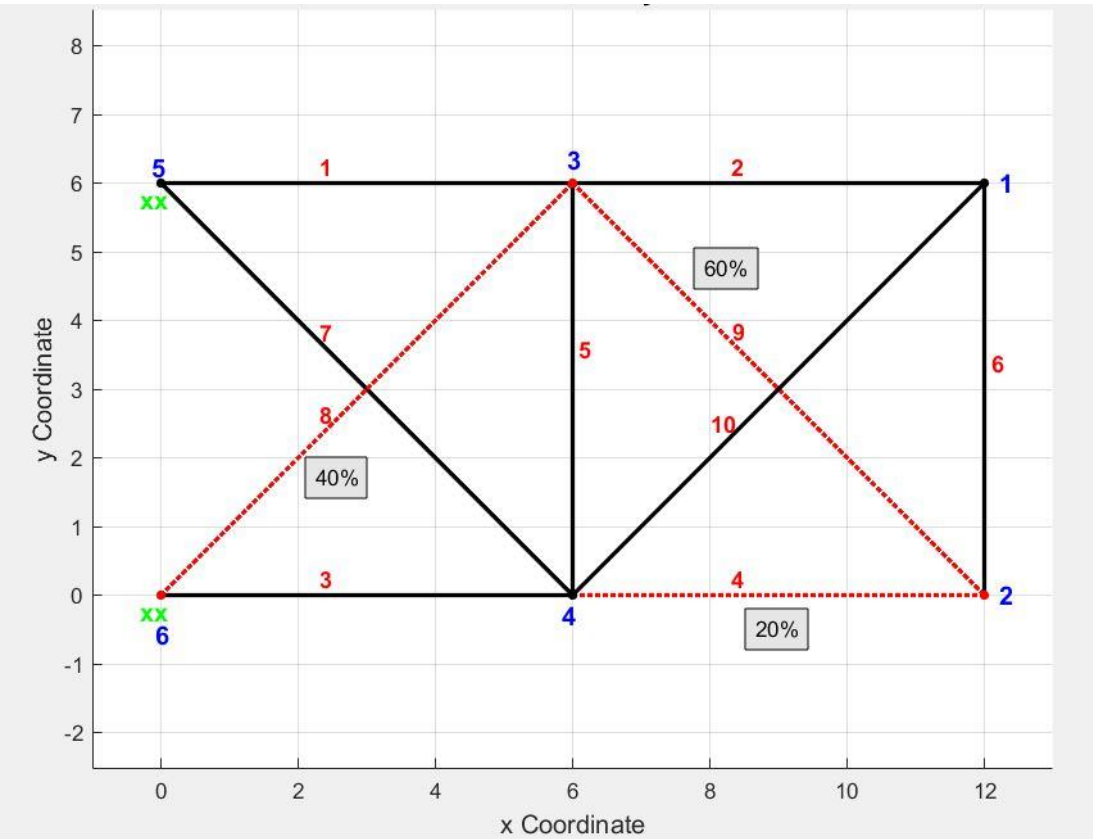

Figure 22 10-bar plane truss with three damaged elements

i) Without the application of noise

a) GA Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 32 Optimization results with GA for three damaged elements using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	33.8	23.2	27.1	53.5	63.2	0	22.7	64.8	78.5	22.4
2 modes	16.7	16.6	16.7	31.9	11.4	17.3	16.2	50.0	66.3	17.3
3 modes	29.8	29.8	29.8	43.8	29.8	30.0	29.8	57.9	71.9	29.8
4 modes	29.6	29.6	29.6	43.6	29.6	29.6	29.6	57.7	71.9	29.6

Mode	Min. objective function value
1	1.01E-05
2	0.000343
3	8.00E-06
4	7.02E-06

b) PSO Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 33 Optimization results with PSO for three damaged elements using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	0	0	0	19.9	0.1	0	0	39.9	59.9	0
2 modes	0	0	0	20.0	0	0	0	40.0	60.0	0
3 modes	0	0	0	20.0	0	0	0	40.0	60.0	0
4 modes	0	0	0	20.0	0	0	0	40.0	60.0	0

Mode	Min. objective function value
1	1.04E-09
2	2.53E-08
3	1.35E-09
4	4.91E-12

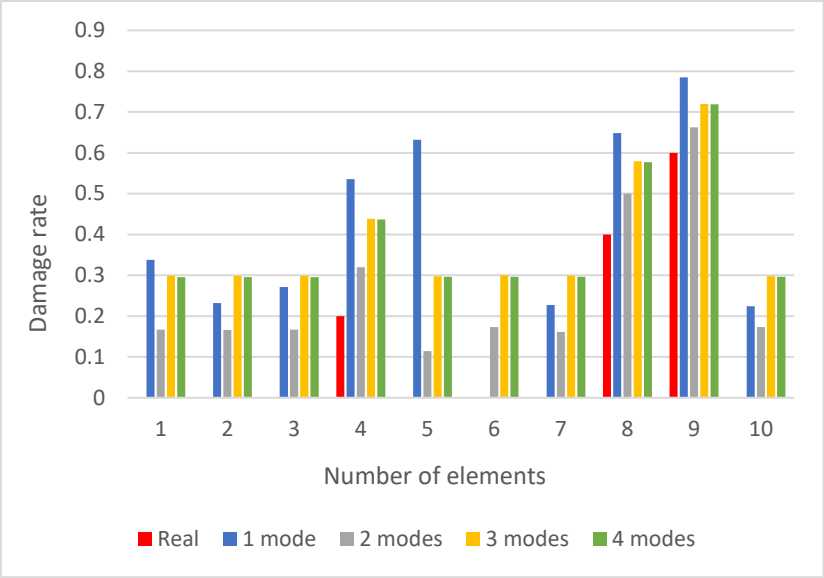

Figure 23 Damage results - Real vs Calculated (GA) using objective function y_1.

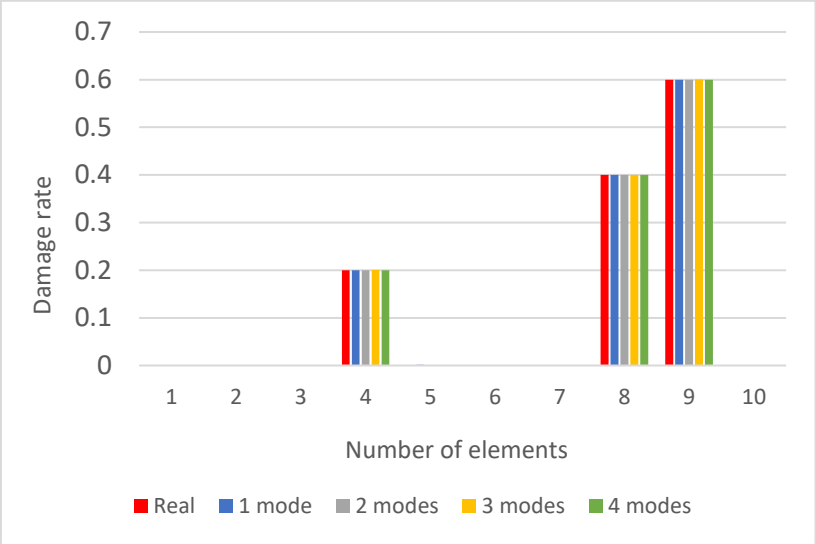

Figure 24 Damage results - Real vs Calculated (PSO) using objective function y_1.

Figure 23 and Figure 24 show the performance of the first objective function (y_1) for the second damage scenario. We can see very clearly that the PSO algorithm is able to identify the damages 100% both in extent and location. The GA algorithm appears to be not able to give good results with the first objective function (y_1), as it demonstrated damage in all elements.

c) GA Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 34 Optimization results in percentage with GA for three damaged elements using objective function y_2.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	2.2	22.2	4.6	12.4	73.9	37.8	13.4	16.8	17.8	40.1
2 modes	1.2	13.9	1.4	0	17.7	2.3	13.1	31.3	49.2	0.7
3 modes	5.6	7.8	0.3	10.8	2.3	6.4	8.9	35.1	41.3	1.6
4 modes	0.4	1.0	0	17.5	6.5	2.2	1.9	39.1	59.6	1.1

Mode	**Min. objective function value**
1	0.0005606
2	0.0043887
3	0.0063679
4	0.0019451

d) PSO Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 35 Optimization results in percentage with PSO for three damaged elements using objective function y_2.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	0	1.7	3.4	0.2	41.9	45.8	1.3	35.7	48.7	13.7
2 modes	0	0	0	7.8	8.3	26.3	0.4	39.6	60.8	0.6
3 modes	0.3	0.3	0	19.8	0.3	3.4	0	40.0	58.6	1.12
4 modes	0	0.2	0	19.7	2.3	0.7	0	39.9	59.9	0

Mode	**Min. objective function value**
1	1.93E-05
2	3.84E-05
3	3.73E-05
4	7.76E-05

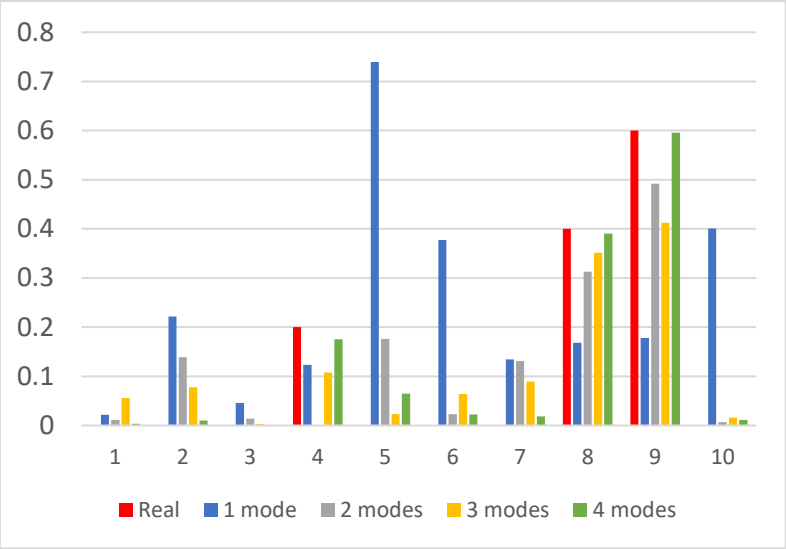

Figure 25 Damage results - Real vs Calculated (GA) using objective function y_2.

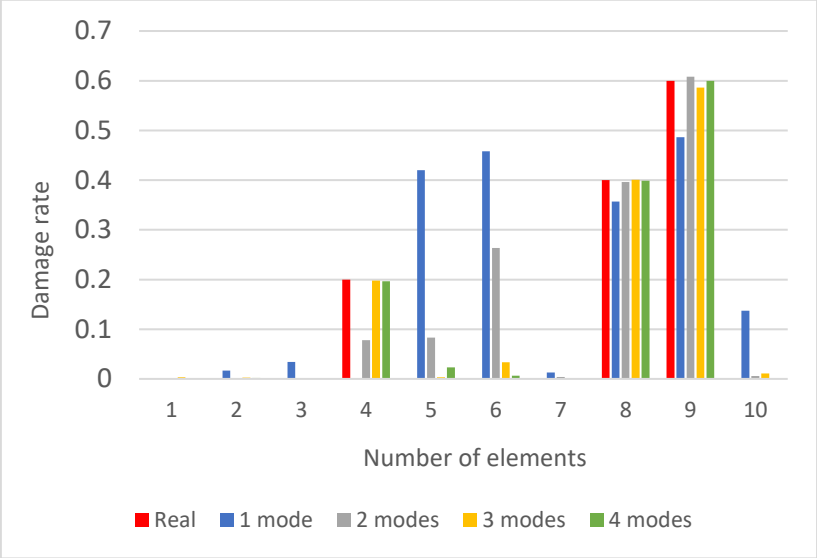

Figure 26 Damage results - Real vs Calculated (PSO) using objective function y_2.

Figure 25 and Figure 26 show the performance of the second objective (y_2) function for the second damage scenario using both GA and PSO algorithms. Again, this objective function exhibits a very good performance. Although both algorithms are able to indicate the position and extent of the damage, especially in the case with 4 known eigenmodes, the PSO exhibits a better performance than the GA for this damage scenario.

e) GA Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 36 Optimization results in percentage with GA for three damaged elements using objective function y_3.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	4.3	19.0	9.7	10.3	18.4	48.9	12.9	39.4	55.3	13.9
2 modes	12.7	3.8	10.9	32.1	27.9	5.6	10.9	49.3	64.7	7.9
3 modes	0	0.7	0.2	19.6	0.3	0.1	0.4	39.6	60.1	0.6
4 modes	0	0	0.7	19.3	0.3	0.5	0.3	39.6	60.3	0.4

Mode	**Min. objective function value**
1	0.036278
2	0.11493
3	0.001189
4	0.003363

f) PSO Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 37 Optimization results in percentage with PSO for three damaged elements using objective function y_3.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	0	1.9	0	17.3	47.9	8.8	0	40.0	58.2	2.3
2 modes	0	0	0	19.9	0.1	0.3	0	40.0	60.0	0
3 modes	0	0	0	20.0	0	0	0	40.0	60.0	0
4 modes	0	0	0	20.0	0.1	0.2	0	40.0	60.0	0

Mode	**Min. objective function value**
1	5.23E-07
2	1.57E-06
3	3.64E-06
4	3.11E-06

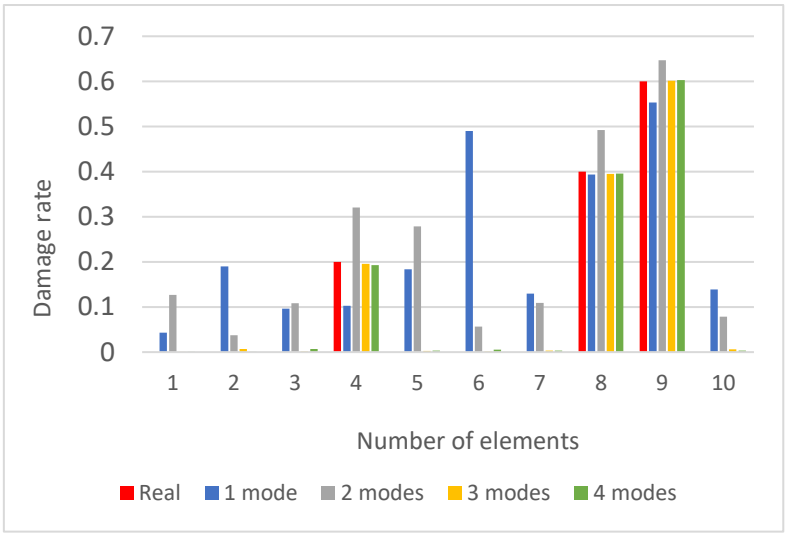

Figure 27 Damage results - Real vs Calculated (GA) using objective function y_3.

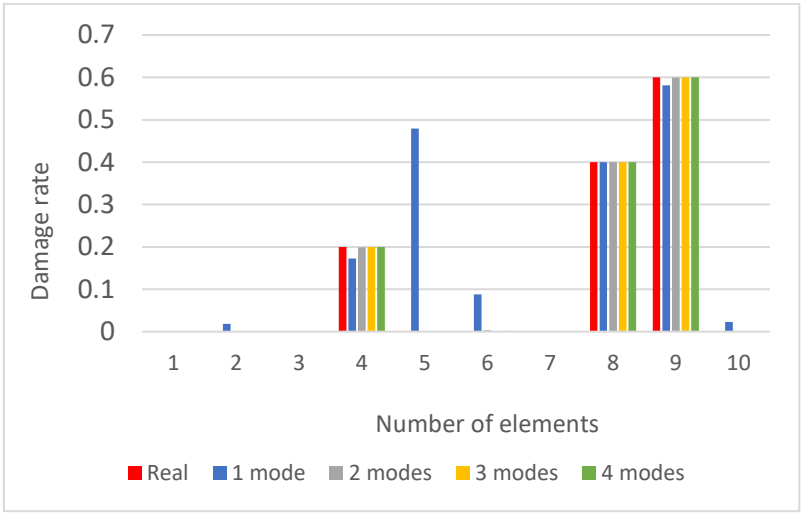

Figure 28 Damage results - Real vs Calculated (PSO) using objective function y_3.

Figure 27 and Figure 28 show the performance of the third objective (y_3) function for the second damage scenario using both algorithms. The results show that both algorithms performed well in identifying the position of the damage as well as the extent of it when using the third objective function. In the cases with 3 and 4 known eigenmodes, the GA algorithm is able to deliver very good results. The PSO algorithm on the other hand, shows an excellent performance in identifying the damages position and extent, especially in the cases of 2, 3 and 4 known eigenmodes.

ii) With applied noise of 5%

a) GA Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 38 Optimization results in percentage with GA for three damaged elements with 5% added noise using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	28.1	38.2	34.7	8.9	0.3	91.4	25.9	53.2	70.4	18.5
2 modes	23.8	30.9	31.9	30.6	38.9	87.3	30.5	57.4	74.2	11.3
3 modes	25.9	31.8	23.4	43.4	42.8	65.9	20.3	55.4	77.9	1.6
4 modes	20.5	24.4	17.9	40.9	31.9	13.5	16.4	52.8	70.5	22.3

Mode	Min. objective function value
1	0.000195
2	0.1173
3	0.42931
4	0.8324

b) PSO Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 39 Optimization results in percentage with PSO for three damaged elements with 5% added noise using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	24.9	25.9	27.2	15.8	27.3	84.3	13.8	54.5	73.9	0.2
2 modes	17.3	26.7	25.2	28.9	32.6	85.1	24.4	53.9	72.5	2.6
3 modes	24.3	30.2	21.7	42.1	41.3	65.1	18.3	54.4	77.5	0
4 modes	11.1	15.5	8.3	33.9	23.8	3.3	6.5	47.2	67.1	13.1

Mode	Min. objective function value
1	1.22E-08
2	0.11408
3	0.42945
4	0.8324

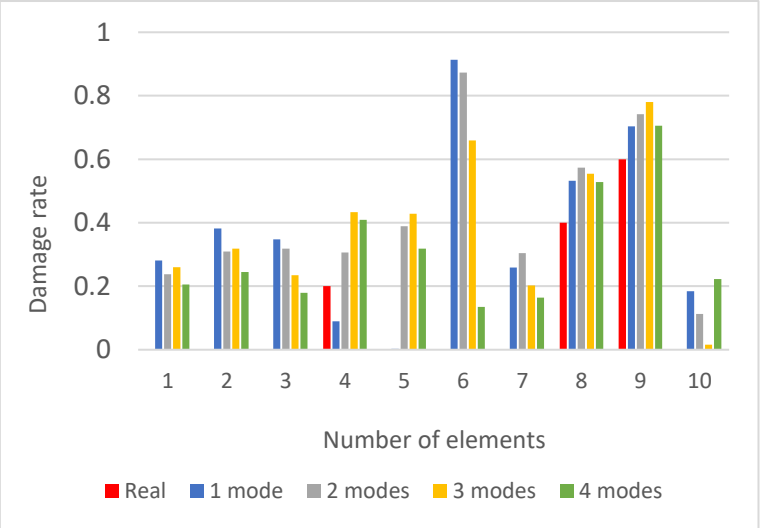

Figure 29 Damage results - Real vs Calculated (GA) using objective function y_1 with 5% noise.

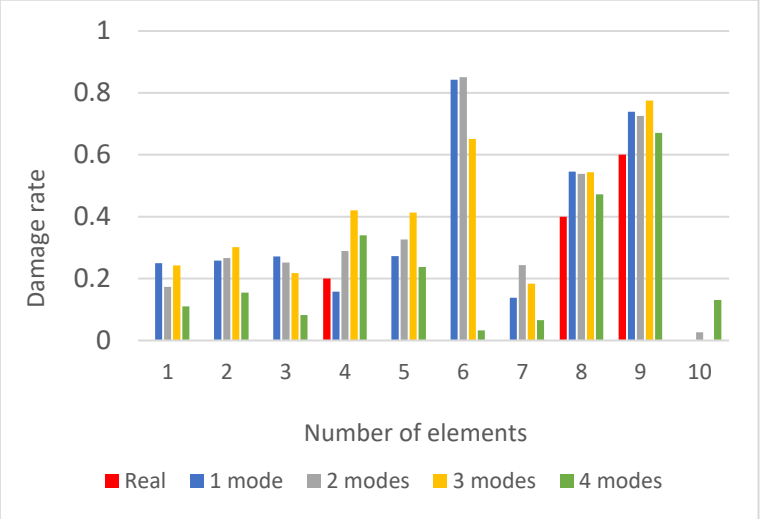

Figure 30 Damage results - Real vs Calculated (PSO) using objective function y_1 with 5% noise.

Figure 29 and Figure 30 show the performance of the first objective function (y_1) for the second damage scenario using both algorithms, when noise of 5% is applied to the mode shapes of the structure. As it is shown in the figures, both algorithms did not exhibit a good performance with the first objective function. They are able to indicate the damage in all three elements, where the real damage is located, but they also show damage existence in all the other elements, even in the case of 4 known eigenmodes.

c) GA Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 40 Optimization results in percentage with GA for three damaged elements with 5% added noise using objective function y_2.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	4.3	0.2	4.4	14.0	5.0	96.9	20.2	17.5	37.4	7.1
2 modes	2.2	11.4	0	0.7	16.4	3.6	10.8	33.3	47.6	6.2
3 modes	0.8	2.3	3.3	11.5	16.9	17.3	1.4	42.5	32.7	14.0
4 modes	0.6	0.6	0.2	17.6	6.4	9.9	1.1	39.6	58.5	3.2

Mode	**Min. objective function value**
1	0.0004542
2	0.0034256
3	0.0095643
4	0.007588

d) PSO Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 41 Optimization results in percentage with PSO for three damaged elements with 5% added noise using objective function y_2.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	1.4	3.4	4.3	0	12.7	90.1	0.2	31.2	54.1	0.4
2 modes	0	2.5	0	0	6.2	57.8	2.2	38.4	59.8	0
3 modes	0.9	1.9	0	18.2	1.6	6.0	1.9	39.0	57.2	0.1
4 modes	0	0	0	19.9	0.2	0.7	0	39.9	59.9	0.1

Mode	**Min. objective function value**
1	5.66E-05
2	0.001069
3	0.002574
4	0.003224

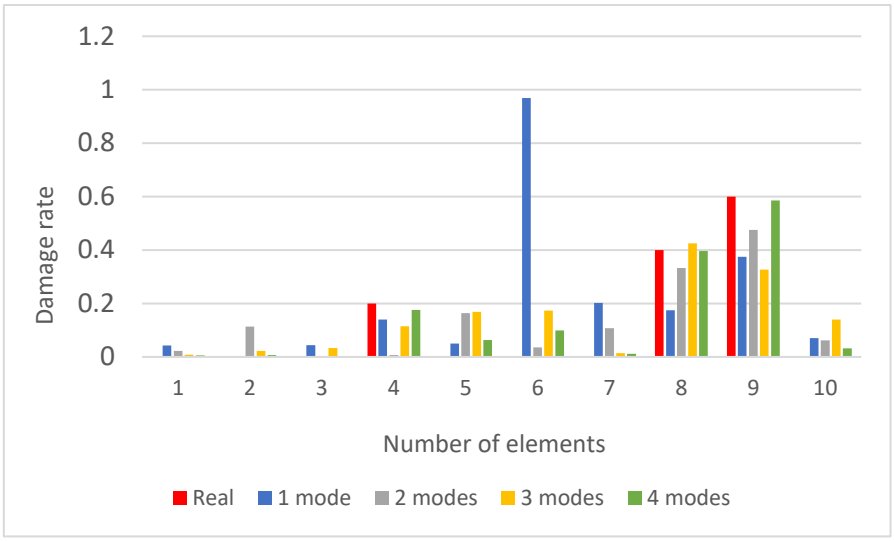

Figure 31 Damage results - Real vs Calculated (GA) using objective function y_2 with 5% noise.

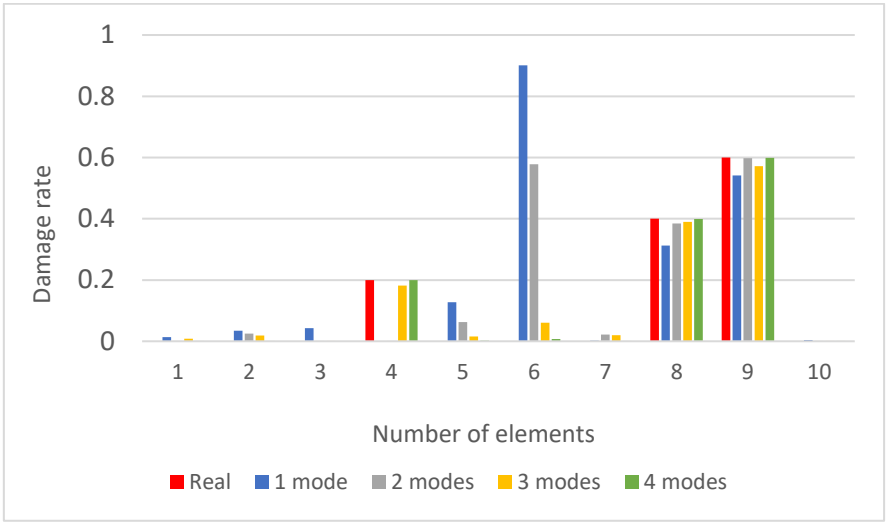

Figure 32 Damage results - Real vs Calculated (PSO) using objective function y_2 with 5% noise.

Figure 31 and Figure 32 show the performance of the second objective function (y_2) for the second damage scenario using both GA and PSO algorithms. The second objective function has exhibited a very good performance in identifying the damage. In this case, the GA, with 4 known eigenmodes shows very good performance, while the PSO shows excellent performance with 3 and 4 known eigenmodes, as it is able to almost identify the exact real damages location and extent.

e) GA Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 42 Optimization results in percentage with GA for three damaged elements with 5% added noise using objective function y_3.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	18.7	28.0	22.8	2.6	0.3	88.9	6.9	52.7	67.1	6.6
2 modes	14.6	22.2	22.7	27.6	45.8	85.6	21.9	53.2	70.5	1.7
3 modes	10.8	18.1	7.6	33.3	29.6	57.8	5.9	47.1	71.4	0.2
4 modes	12.0	16.3	8.8	35.2	24.7	4.2	6.8	48.5	66.9	13.9

Mode	**Min. objective function value**
1	0.093357
2	0.22758
3	0.51034
4	0.87177

f) PSO Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 43 Optimization results in percentage with PSO for three damaged elements with 5% added noise using objective function y_3.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	0.8	15.5	8.1	0	0.3	89.9	2.8	34.3	56.7	0
2 modes	2.7	9.9	10.6	16.6	37.5	83.4	10.8	44.9	66.5	0
3 modes	11.3	18.5	7.9	33.5	29.9	58.0	6.4	47.4	71.7	0
4 modes	2.9	7.6	0.3	28.1	17.7	0	0	42.6	63.7	5.3

Mode	**Min. objective function value**
1	0.027067
2	0.16903
3	0.50951
4	0.84129

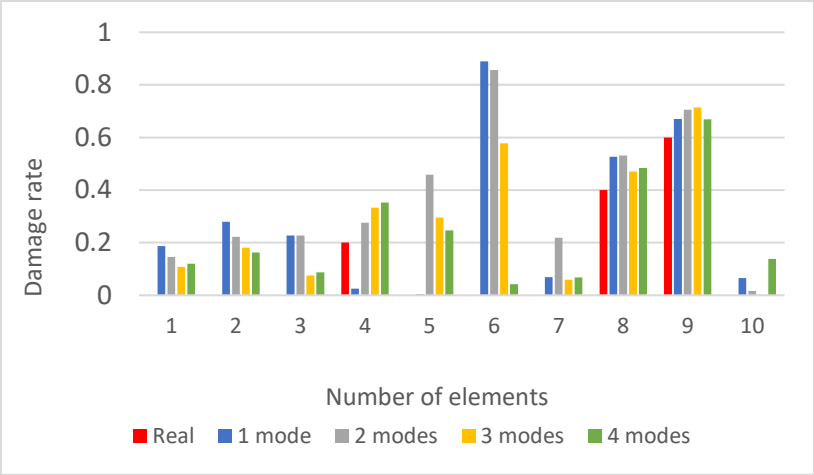

Figure 33 Damage results - Real vs Calculated (GA) using objective function y_3 with 5% noise.

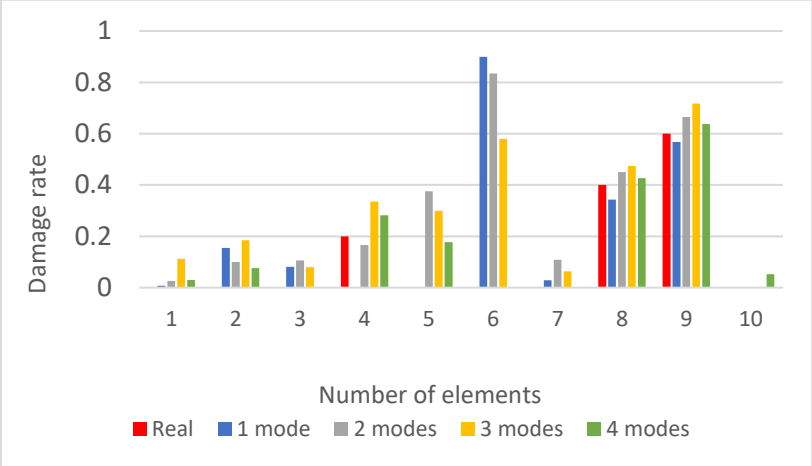

Figure 34 Damage results - Real vs Calculated (PSO) using objective function y_3 with 5% noise.

Figure 33 and Figure 34 show the performance of the third objective function (y_3) for the second damage scenario using both algorithms. Here, we clearly see that the application of noise in the calculated data influences its values. Both algorithms are able to indicate the position of the real damage but not the extent. They also show various damage rates in the healthy (undamaged) elements.

iii) With applied noise of 10%

a) GA Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 44 Optimization results in percentage with GA for three damaged elements with 10% added noise using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	73.9	56.3	65.8	83.1	0.6	89.7	43.2	92.1	99.6	5.2
2 modes	28.2	45.6	34.9	48.8	1.1	89.6	34.2	62.3	80.4	4.0
3 modes	45.2	54.1	42.9	62.4	66.2	81.0	37.6	67.3	87.2	0.1
4 modes	27.3	30.6	22.8	51.2	53.6	69.9	22.3	56.9	62.6	58.1

Mode	Min. objective function value
1	0.000479
2	0.6674
3	1.773
4	2.9549

b) PSO Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 45 Optimization results in percentage with PSO for three damaged elements with 10% added noise using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	43.9	47.7	47.5	0	2.8	94.3	26.6	65.2	80.8	0
2 modes	24.6	43.1	35.0	2.1	5.7	89.9	30.7	59.3	79.4	0
3 modes	43.8	52.8	40.8	61.1	64.9	80.2	35.4	66.4	86.8	0
4 modes	9.8	17.7	3.6	38.1	33.0	8.1	0	46.9	68.5	11.9

Mode	Min. objective function value
1	0.00213
2	0.69929
3	1.7752
4	3.3983

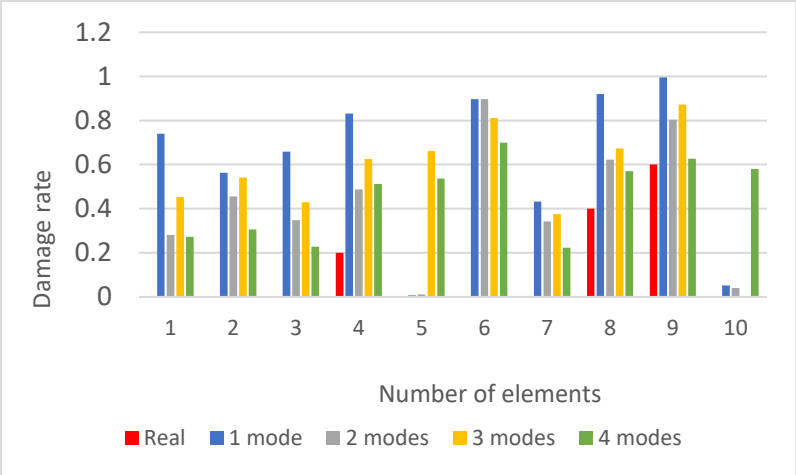

Figure 35 Damage results - Real vs Calculated (GA) using objective function y_1 with 10% noise.

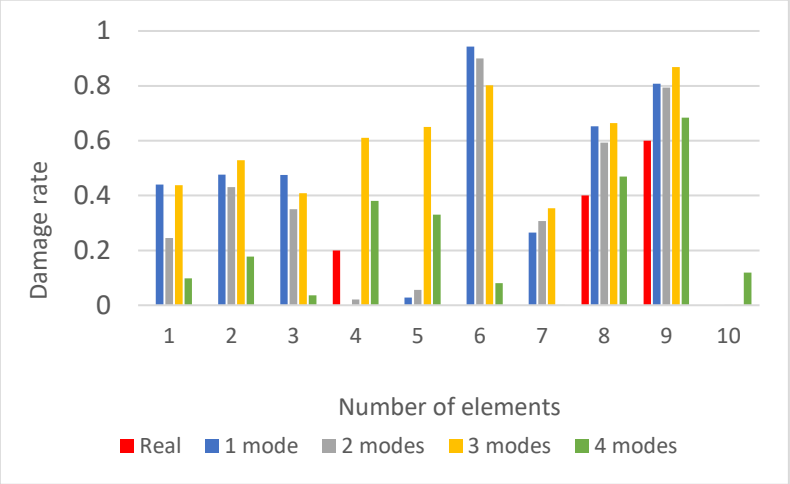

Figure 36 Damage results - Real vs Calculated (PSO) using objective function y_1 with 10% noise.

Figure 35 and Figure 36 show the performance of the first objective (y_1) function for the second damage scenario using both algorithms, when a noise of 10% is applied to the mode shapes of the structure. With a high noise ratio, we see that both algorithms are not exhibiting a good performance with the first objective function. Even though they are able to detect the location of the real damage, they also show damage is present in the healthy elements.

c) GA Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 46 Optimization results in percentage with GA for three damaged elements with 10% added noise using objective function y_2.

	1	2	3	4	5	6	7	8	9	10
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	2.1	3.1	2.5	17.4	59.6	96.4	7.8	15.8	55.9	1.6
2 modes	0.7	4.1	2.6	11.2	8.1	40.2	3.8	39.5	46.4	2.2
3 modes	0.6	0	0.3	18.5	5.6	0	0	40.8	54.5	4.3
4 modes	0.2	0.2	0	18.3	8.6	2.8	0.2	40.1	58.7	2.7

Mode	Min. objective function value
1	0.00057
2	0.006861
3	0.009642
4	0.015329

d) PSO Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 47 Optimization results in percentage with PSO for three damaged elements with 10% added noise using objective function y_2.

	1	2	3	4	5	6	7	8	9	10
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	0	7.2	5.7	0.6	44.0	97.6	0	25.8	53.1	0.5
2 modes	0	5.7	0	0	0.7	61.5	3.9	36.8	58.8	0
3 modes	0	1.3	0	19.7	0	2.4	0.2	39.9	57.6	2.8
4 modes	0	0.1	0	19.9	0	0	0	39.9	60.0	0

Mode	Min. objective function value
1	3.17E-03
2	0.004813
3	0.009082
4	0.012921

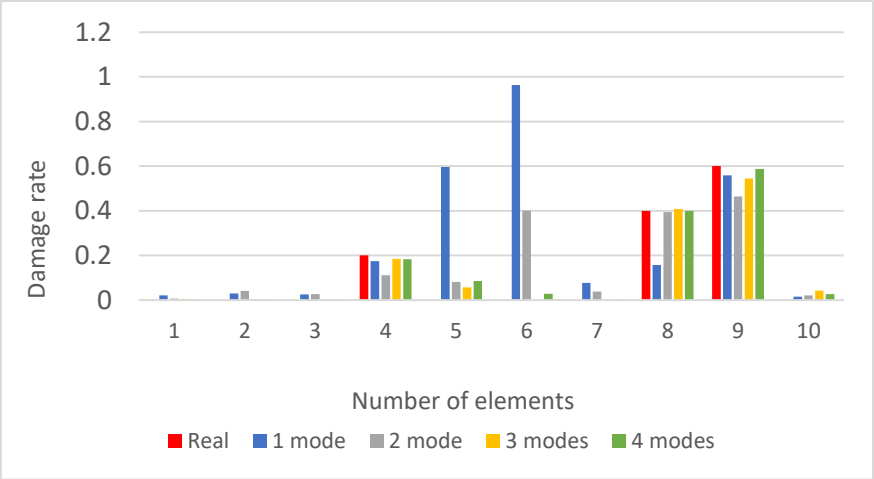

Figure 37 Damage results - Real vs Calculated (GA) using objective function y_2 with 10% noise.

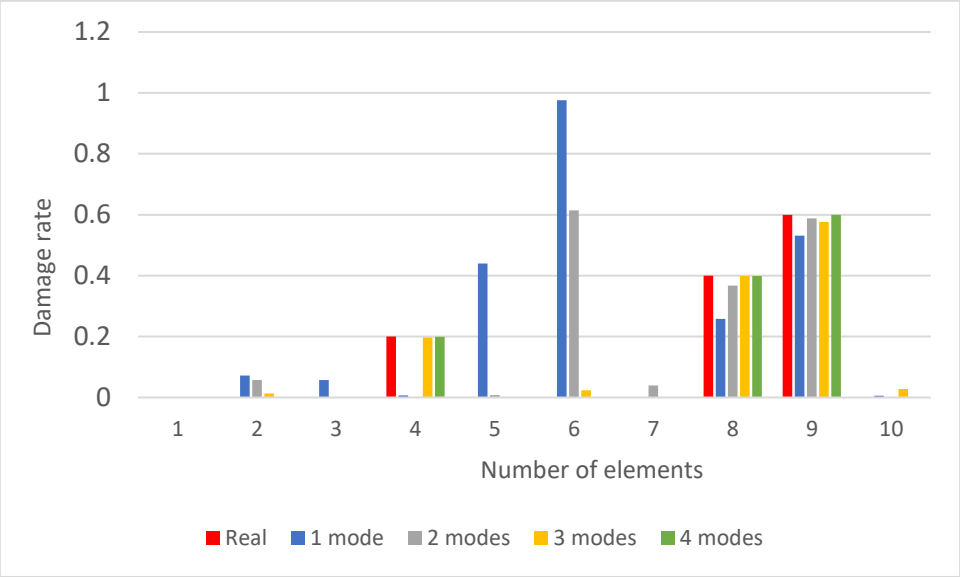

Figure 38 Damage results - Real vs Calculated (PSO) using objective function y_2 with 10% noise.

Figure 37 and Figure 38 show the performance of the second objective function (y_2) for the second damage scenario for both algorithms, when noise of 10% is applied to the mode shapes of the structure. Once again, we can observe that when using a high noise rate, the second objective function performs better than the others in determining the position and degree of the damage. The PSO method surpasses the GA approach in determining the precise damage rates in the case with more known eigenmodes.

e) GA Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 48 Optimization results in percentage with GA for three damaged elements with 10% added noise using objective function y_3.

	1	2	3	4	5	6	7	8	9	10
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	24.4	42.9	34.1	0.1	3.1	96.4	18.7	49.7	69.6	0.4
2 modes	42.3	58.3	46.4	58.0	2.7	91.7	46.3	70.4	85.0	6.1
3 modes	38.2	48.1	34.2	57.0	60.9	78.0	29.6	63.5	85.3	0.2
4 modes	13.14	20.9	7.1	40.4	35.5	12.2	4.5	49.3	69.4	15.1

Mode	Min. objective function value
1	0.14069
2	0.77582
3	1.8796
4	3.4139

f) PSO Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 49 Optimization results in percentage with PSO for three damaged elements with 10% added noise using objective function y_3.

	1	2	3	4	5	6	7	8	9	10
Real	0	0	0	20.0	0	0	0	40.0	60.0	0
1 mode	18.0	33.9	25.2	0	0	96.7	10.4	44.1	65.2	0
2 modes	20.3	41.2	27.9	27.2	0	89.2	25.4	58.7	79.1	0
3 modes	38.2	48.4	34.7	57.4	61.1	78.2	29.8	63.5	85.2	0
4 modes	8.4	16.7	2.3	37.0	32.2	6.6	0	46.4	67.9	10.8

Mode	Min. objective function value
1	0.12254
2	0.73541
3	1.8785
4	3.4085

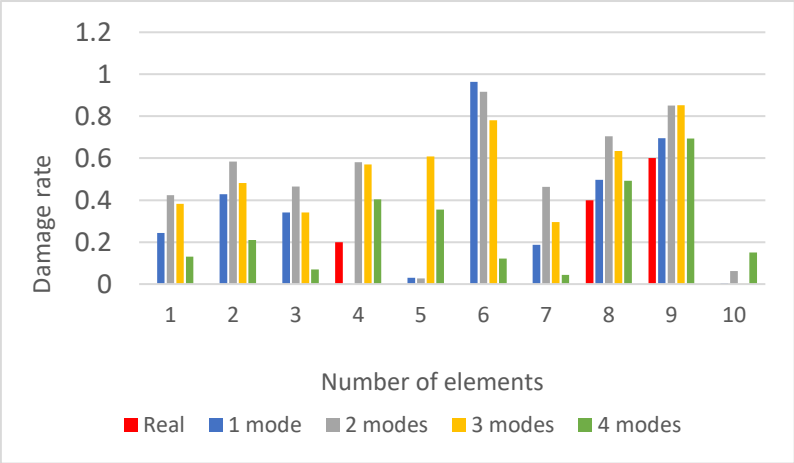

Figure 39 Damage results - Real vs Calculated (GA) using objective function y_3 with 10% noise.

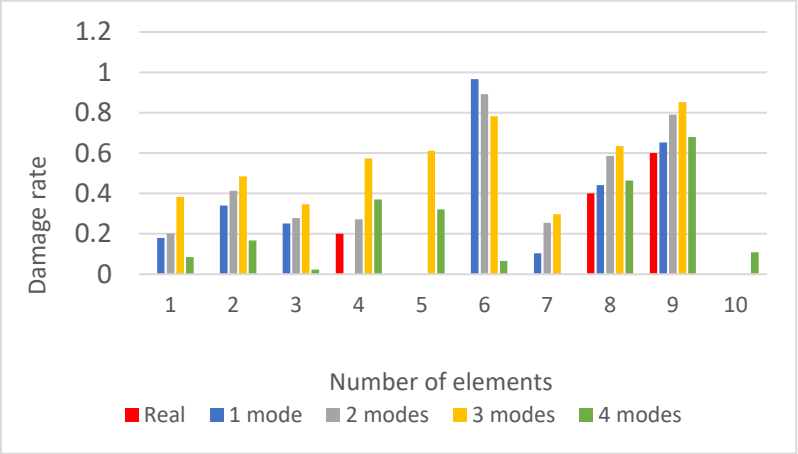

Figure 40 Damage results - Real vs Calculated (PSO) using objective function y_3 with 10% noise.

Figure 39 and Figure 40 present the performance of the third objective function (y_3) for the second damage case, when a noise of 10% is applied to the mode shapes of the structure. The third objective function does not perform well in pinpointing the degree and location of the damage in the structure when such a high noise level is present. This is true for both optimization techniques.

6.2.3. Damage scenario 3

The last damage scenario is going to be a uniform damage of 20% to all elements of the truss. Similar to the two previous scenarios, multiple tests with the three objective functions using equations (24), (25) and (26) are carried out within this damage scenario as well. The first set of tests were carried out without the application of noise to each mode shape. The second set were tested with the application of 5% & 10% noise. Additionally, tests were carried out for 1, 2, 3 and 4 known eigenmodes.

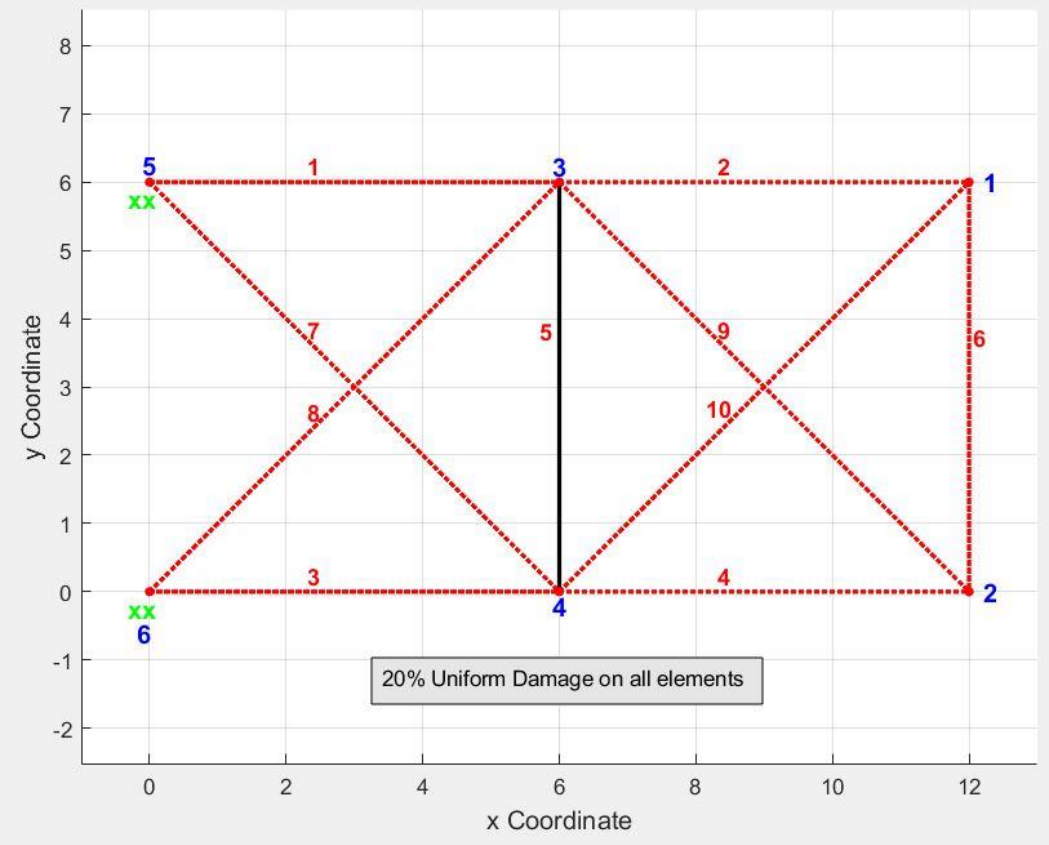

Figure 41 20% Uniform damage for 10-bar plane truss

i) Without the application of noise

a) GA Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 50 Optimization results in percentage with GA for uniform damage using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	41.6	40.9	39.9	40.9	1.0	12.2	38.5	43.1	40.6	41.2
2 modes	39.7	39.4	39.8	39.1	40.3	43.9	39.9	39.7	39.8	39.7
3 modes	28.5	28.4	28.5	28.5	28.5	28.6	28.5	28.5	28.5	28.4
4 modes	45.8	45.8	45.8	45.8	45.8	45.9	45.8	45.8	45.8	45.9

Mode	Min. objective function value
1	0.000456
2	0.000729
3	1.48E-06
4	4.64E-06

b) PSO Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 51 Optimization results in percentage with PSO for uniform damage using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	0	0	0	0	46.1	43.8	0	0	0	0
2 modes	0	0	0	0	0	0	0	0	0	0
3 modes	0	0	0	0	0	0	0	0	0	0
4 modes	0	0	0	0	0	0	0	0	0	0

Mode	Min. objective function value
1	-2.22E-13
2	1.11E-14
3	2.22E-14
4	-1.55E-13

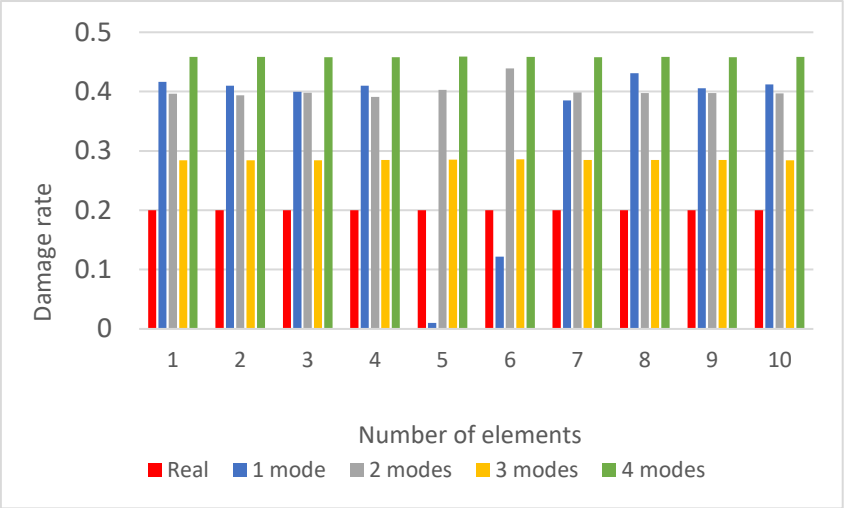

Figure 42 Damage results - Real vs Calculated (GA) using objective function y_1.

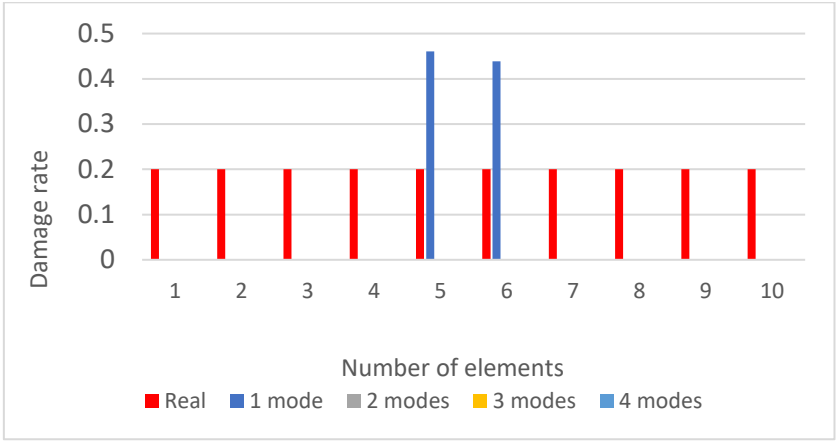

Figure 43 Damage results - Real vs Calculated (PSO) using objective function y_1.

Figure 42 and Figure 43 shows the performance of the first objective function (y_1) for the third and last damage scenario (uniform damage), with both algorithms. The case where uniform damage occurs seems to be the most challenging for the first objective function. Using this objective function, the PSO algorithm entirely fails to pinpoint the location or degree of damage. On the other hand, the GA recognizes that there is uniform damage across the structure but is unable to determine its severity.

c) GA Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 52 Optimization results in percentage with GA for uniform damage using objective function y_2.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	16.6	25.2	15.5	31.6	37.7	11.8	34.9	15.8	16.5	27.1
2 modes	20.7	5.8	20.1	20.5	32.2	57.1	9.0	24.3	9.0	37.5
3 modes	26.6	16.6	13.3	29.6	5.5	1.8	16.2	20.6	16.0	22.3
4 modes	17.1	17.2	22.8	27.3	19.9	7.8	11.7	25.8	19.5	19.8

Mode	**Min. objective function value**
1	0.0001717
2	0.0018529
3	0.0025471
4	0.0026031

d) PSO Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 53 Optimization results in percentage with PSO for uniform damage using objective function y_2.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	19.7	19.8	21.6	8.6	70.9	51.2	15.2	23.7	31.8	5.6
2 modes	24.9	14.0	21.9	20.1	14.2	30.8	9.7	14.3	5.7	26.2
3 modes	24.6	23.6	17.8	18.5	8.6	23.2	24.4	15.1	10.6	11.9
4 modes	19.3	19.9	20.7	19.8	21.4	9.6	19.4	20.7	17.4	22.5

Mode	**Min. objective function value**
1	4.31-05
2	0.00058
3	0.001204
4	0.000511

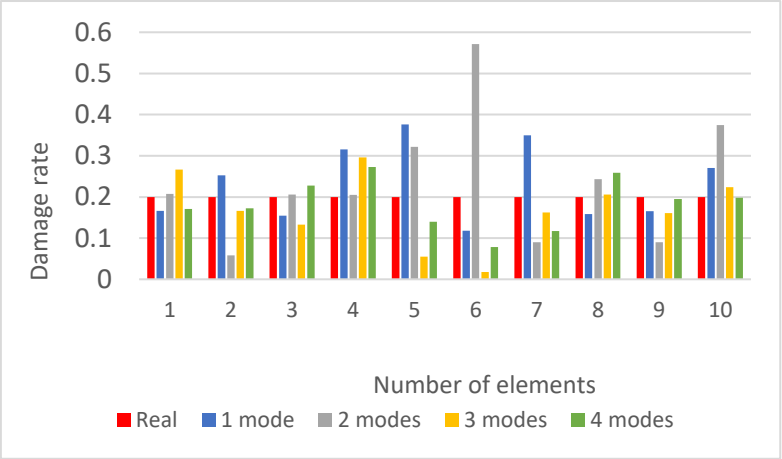

Figure 44 Damage results - Real vs Calculated (GA) using objective function y_2.

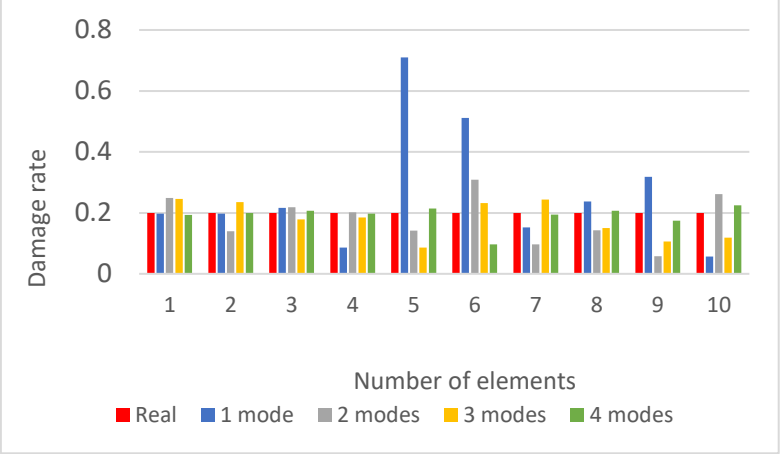

Figure 45 Damage results - Real vs Calculated (PSO) using objective function y_2.

Figure 44 and Figure 45 show the performance of the second objective function (y_2) for the last damage scenario (uniform damage) using both algorithms. Both algorithms show a very good performance in identifying the damage in the structure using the second objective function. When we take all 4 known eigenmodes into consideration, the PSO algorithm performs better in giving a good indication of the damage extent than GA does.

e) GA Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 54 Optimization results in percentage with GA for uniform damage using objective function y_3.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	20.4	17.5	19.5	22.7	2.3	0.5	18.9	20.5	23.0	17.4
2 modes	33.4	31.2	34.3	25.3	33.0	36.0	36.2	32.2	32.5	32.7
3 modes	20.3	19.4	19.7	19.9	20.8	21.4	20.0	20.3	20.0	19.8
4 modes	20.0	20.4	20.0	20.3	19.1	17.5	19.7	19.9	19.7	20.2

Mode	**Min. objective function value**
1	0.000765
2	0.1611
3	0.000496
4	0.001

f) PSO Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 55 Optimization results in percentage with PSO for uniform damage using objective function y_3.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	20	19.8	20.0	20.2	6.2	3.2	20.0	20.0	20.4	19.7
2 modes	20	20.	20.0	20.1	20.1	18.5	20.0	20.0	20.1	20.2
3 modes	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
4 modes	20	20.1	20.0	20.0	20.0	19.8	20.0	19.9	19.9	20.1

Mode	**Min. objective function value**
1	7.98E-06
2	7.62E-06
3	1.07E-05
4	1.99E-05

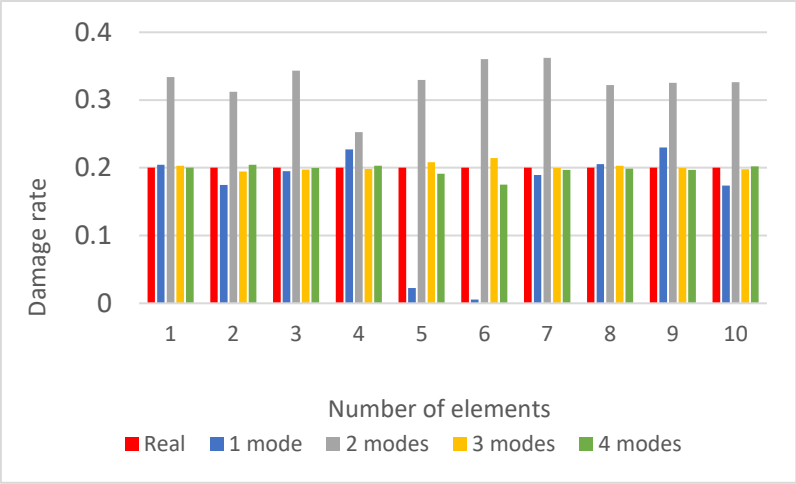

Figure 46 Damage results - Real vs Calculated (GA) using objective function y_3.

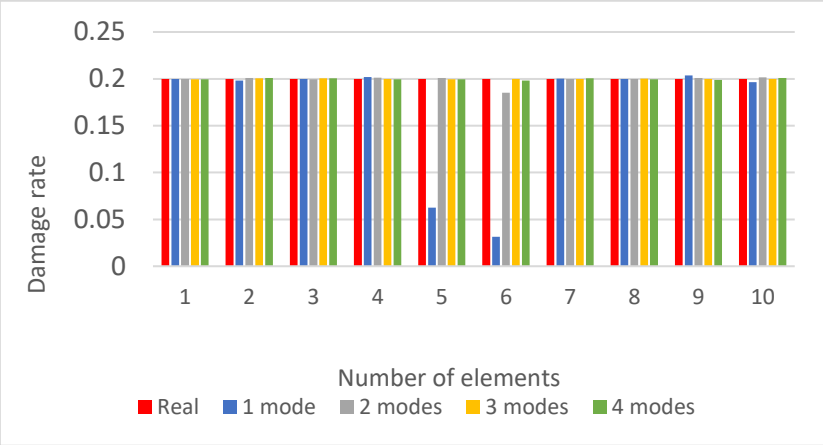

Figure 47 Damage results - Real vs Calculated (PSO) using objective function y_3.

Figure 46 and Figure 47 show the performance of the third objective function (y_3) for the last damage scenario (uniform damage) using both algorithms. As can be seen in both figures, the third objective function is able to identify the damage in all the elements in both algorithms. As for the extent of the damage, again, both algorithms show an almost 100% match with more known eigenmodes.

ii) With applied noise of 5%

a) GA Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 56 Optimization results in percentage with GA for uniform damage with 5% added noise using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	38.2	31.6	36.4	24.9	42.4	93.1	20.5	39.9	54.3	0.1
2 modes	55.2	45.8	57.3	42.7	64.4	84.4	50.8	49.8	76.7	8.4
3 modes	45.9	48.6	43.3	41.1	50.6	70.9	37.4	46.1	63.3	14.1
4 modes	30.0	34.4	29.7	28.4	31.9	55.6	23.6	30.9	36.2	38.6

Mode	Min. objective function value
1	0.000297
2	0.47312
3	0.63835
4	0.5887

b) PSO Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 57 Optimization results in percentage with PSO for uniform damage with 5% added noise using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	28.	30.9	30.8	3.7	0	99.9	17.6	25.6	40.8	0
2 modes	31.2	15.2	30.1	14.5	41.4	71.7	23.7	21.9	56.4	0
3 modes	33.6	36.7	29.8	28.5	38.6	63.5	22.6	33.9	53.4	0.1
4 modes	7.4	13.9	6.8	4.9	8.4	38.9	0	8.7	11.5	15.3

Mode	Min. objective function value
1	2.84E-06
2	0.54877
3	0.64063
4	0.58923

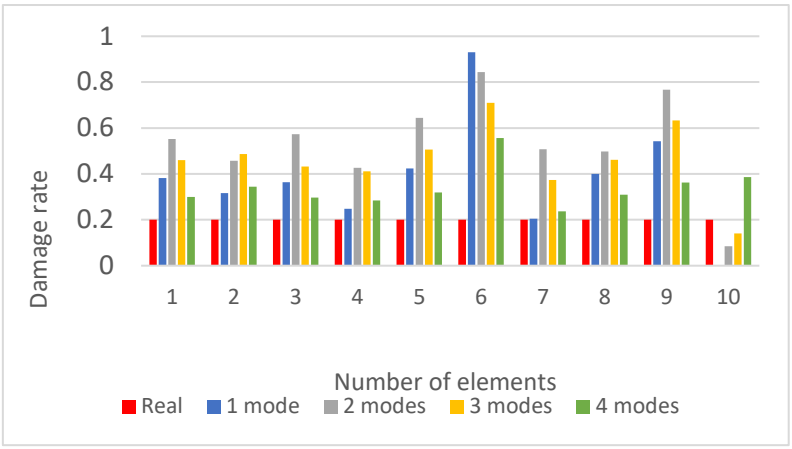

Figure 48 Damage results - Real vs Calculated (GA) using objective function y_1 with 5% noise.

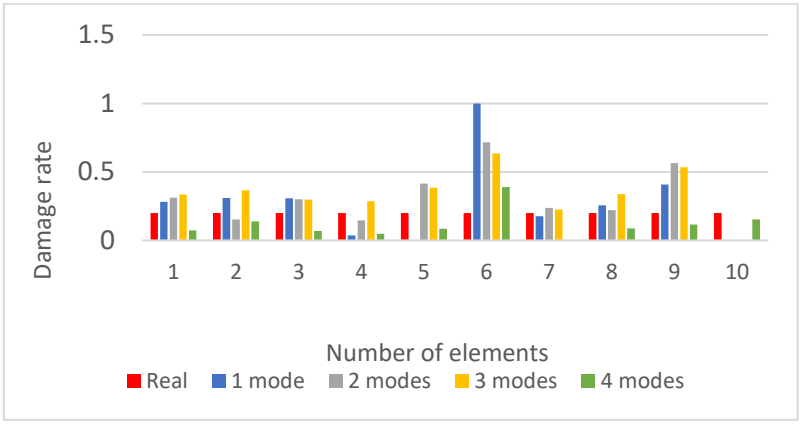

Figure 49 Damage results - Real vs Calculated (PSO) using objective function y_1 with 5% noise.

Figure 48 and Figure 49 show the performance of the first objective function (y_1) for the last damage scenario (uniform damage), when noise of 5% is applied to the mode shapes of the structure. The results demonstrate that the objective function predicts that there is damage in all elements, however, the extent of the damage is far from the real damage, even in the case of 4 known eigenmodes. This shows that the first objective function exhibits poor performance in the case of uniform damage in a structure.

c) GA Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 58 Optimization results in percentage with GA for uniform damage with 5% added noise using objective function y_2.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	23.3	14.9	13.5	18.5	45.1	97.5	4.7	23.5	42.8	27.4
2 modes	20.0	20.1	25.3	15.5	16.1	35.7	11.6	13.1	15.0	24.4
3 modes	20.7	30.2	11.2	27.3	12.4	28.5	19.9	11.1	36.6	36.8
4 modes	14.5	22.7	24.9	20.5	16.9	1.8	17.5	21.1	26.4	12.5

Mode	**Min. objective function value**
1	0.000444
2	0.001642
3	0.006882
4	0.0070701

d) PSO Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 59 Optimization results in percentage with PSO for unform damage with 5% added noise using objective function y_2.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	24.6	10.5	21.9	9.6	47.9	99.9	8.9	19.7	28.2	1.1
2 modes	19.9	27.7	26.6	18.4	6.1	0.8	4.7	0.9	21.4	31.2
3 modes	20.8	20.3	22.8	19.6	9.1	11.7	19.3	21.2	0.9	14.7
4 modes	16.6	17.7	23.0	25.1	17.9	21.4	13.0	25.7	19.1	20.6

Mode	**Min. objective function value**
1	5.13E-05
2	0.001731
3	0.003113
4	0.004271

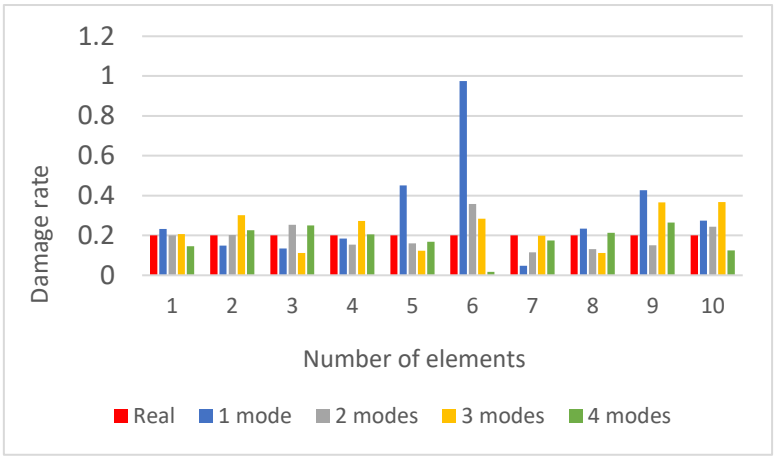

Figure 50 Damage results - Real vs Calculated (GA) using objective function y_2 with 5% noise.

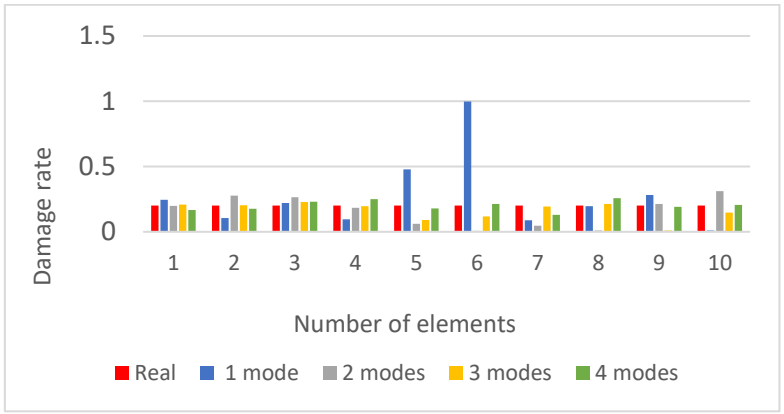

Figure 51 Damage results - Real vs Calculated (PSO) using objective function y_2 with 5% noise.

Figure 50 and Figure 51 show the performance of the second objective function (y_2) for the last damage scenario (uniform damage) when a noise of 5% is applied to the mode shapes of the structure. The second objective function shows a highly excellent performance in identifying a uniform damage in the structure with both algorithms. Once again, the pattern is the same. The PSO algorithm in the second objective function shows a better performance than the GA.

e) GA Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 60 Optimization results in percentage with GA for uniform damage with 5% added noise using objective function y_3.

	1	2	3	4	5	6	7	8	9	10
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	25.6	25.7	24.5	1.8	7.2	99.5	6.6	27.0	37.2	0.5
2 modes	37.8	26.6	39.9	21.9	49.4	77.1	32.4	30.2	64.2	0.3
3 modes	28.6	31.2	26.2	26.0	34.8	60.4	18.4	27.5	48.5	0.2
4 modes	17.9	21.7	18.5	18.3	20.7	45.2	11.5	18.5	21.4	25.8

Mode	Min. objective function value
1	0.017517
2	0.54062
3	0.68495
4	0.61174

f) PSO Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 61 Optimization results in percentage with PSO for uniform damage with 5% added noise using objective function y_3.

	1	2	3	4	5	6	7	8	9	10
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	21.6	22.1	23.8	0	52.3	99.9	10.5	18.8	35.5	0
2 modes	25.3	14.1	26.1	9.2	32.6	67.4	19.8	16.0	51.3	0.2
3 modes	24.2	27.6	21.3	21.4	29.4	56.0	13.5	23.4	43.5	0
4 modes	15.4	20.0	16.0	15.9	17.3	43.8	8.5	16.1	19.1	22.8

Mode	Min. objective function value
1	0.007505
2	0.57641
3	0.67788
4	0.61634

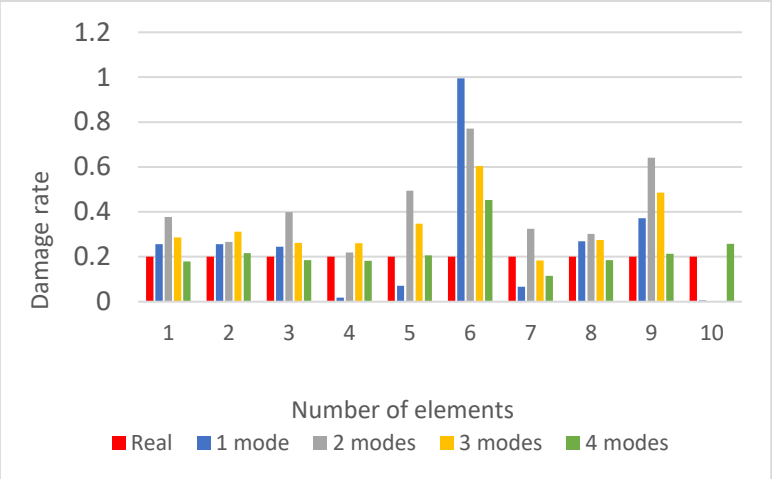

Figure 52 Damage results - Real vs Calculated (GA) using objective function y_3 with 5% noise.

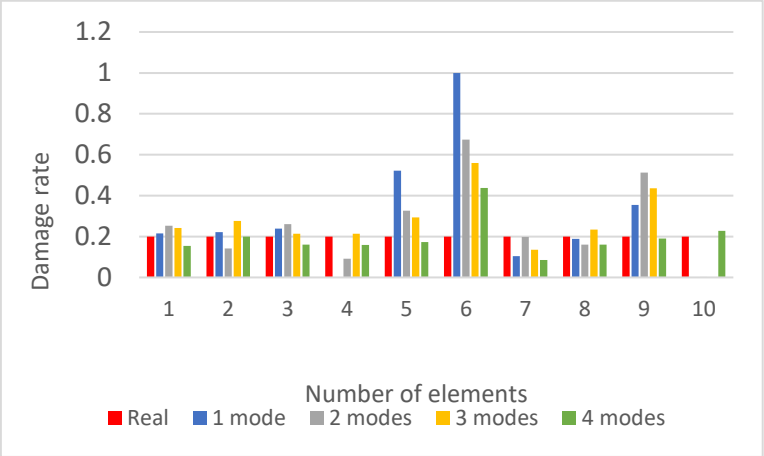

Figure 53 Damage results - Real vs Calculated (PSO) using objective function y_3 with 5% noise.

Figure 52 and Figure 53 show the performance of the third objective function (y_3) for the last damage scenario (uniform damage), when a noise of 5% is applied to the mode shapes of the structure. With both algorithms, the objective function performs well in identifying damage to every component of the structure. Its accuracy for this damage scenario is also considered good, especially in the case with 4 known eigenmodes.

iii) With applied noise of 10%

a) GA Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 62 Optimization results in percentage with GA for uniform damage with 10% added noise using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	59.9	59.9	60.6	26.4	27.5	99.4	42.3	58.2	71.4	2.0
2 modes	68.4	61.4	72.0	54.4	76.4	92.1	64.1	63.6	90.6	1.2
3 modes	62.5	65.5	60.6	59.2	70.4	85.4	50.9	62.9	81.5	0.2
4 modes	30.6	41.4	28.8	27.6	32.5	61.1	16.5	32.3	43.6	47.2

Mode	Min. objective function value
1	0.00263
2	1.0317
3	2.2216
4	2.4167

b) PSO Optimization with the MACFLEX criterion: $y_1 = 100 * (1 - MACFLEX)$

Table 63 Optimization results in percentage with PSO for uniform damage with 10% added noise using objective function y_1.

	1	2	3	4	5	6	7	8	9	10
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	57.5	59.7	58.4	14.9	2.3	99.9	37.6	56.8	69.4	0
2 modes	59.9	48.2	63.7	44.2	71.8	90.3	51.2	54.7	85.9	0
3 modes	59.9	64.3	58.2	56.6	68.9	84.4	48.6	61.2	80.5	0
4 modes	14.7	27.3	13.9	12.3	13.7	51.2	0	16.0	28.8	33.1

Mode	Min. objective function value
1	0.000718
2	1.1473
3	2.2331
4	2.4194

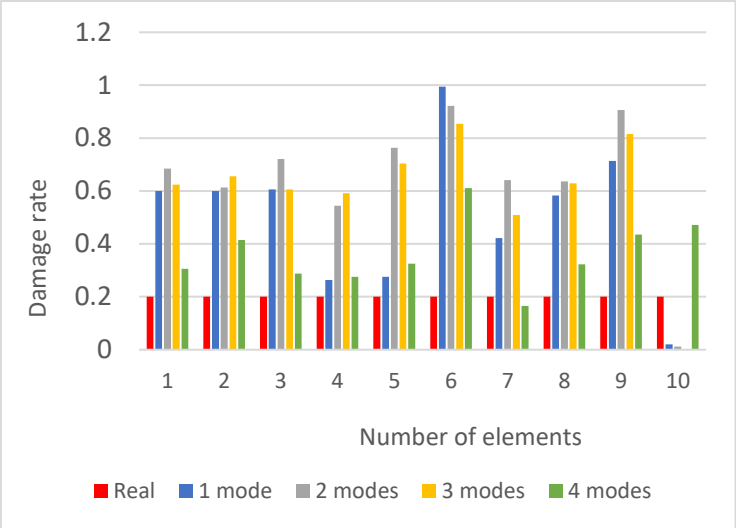

Figure 54 Damage results - Real vs Calculated (GA) using objective function y_1 with 10% noise.

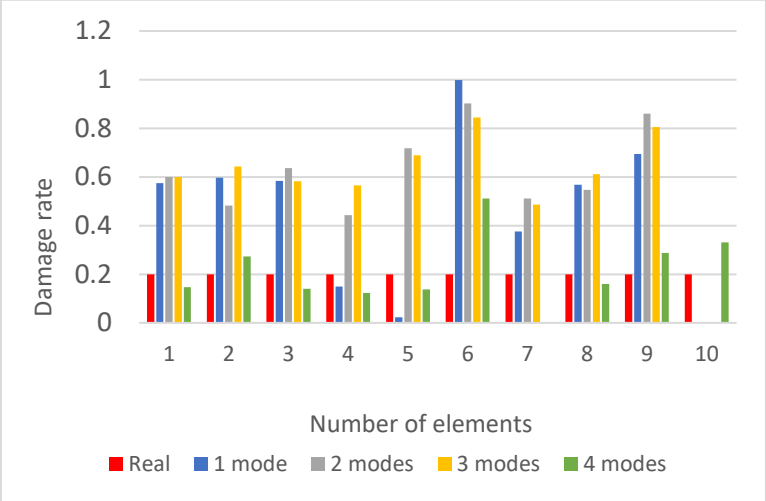

Figure 55 Damage results - Real vs Calculated (PSO) using objective function y_1 with 10% noise.

Figure 54 and Figure 55 show the performance of the first objective function (y_1) for the last damage scenario (uniform damage), when a noise of 10% is applied to the mode shapes of the structure. As was evident in the damage cases above, the first objective function performs poorly when a large noise rate is applied. Both algorithms show damage across the entire structure, but they are not very accurate.

c) GA Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 64 Optimization results in percentage with GA for uniform damage and 10% added noise using objective function y_2.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	23.7	1.6	12.3	44.3	65.4	93.9	0.7	24.0	53.8	4.3
2 modes	26.1	15.5	26.4	14.6	10.2	39.9	2.2	7.3	3.1	17.8
3 modes	13.4	20.6	25.5	20.9	20.1	19.9	14.7	24.8	21.1	18.7
4 modes	21.4	12.8	18.4	23.0	24.5	24.8	16.4	24.7	19.5	21.1

Mode	**Min. objective function value**
1	0.00085
2	0.007704
3	0.022238
4	0.01576

d) PSO Optimization with the MTMAC criterion: $y_2 = 1 - MTMAC$

Table 65 Optimization results in percentage with PSO for uniform damage and 10% added noise using objective function y_2.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	21.1	5.5	25.3	2.7	99.8	99.9	6.8	8.8	43.3	0.5
2 modes	17.7	32.3	22.5	7.6	12.4	21.0	23.1	6.5	18.7	36.9
3 modes	8.9	30.0	28.2	16.4	14.6	0.4	16.6	20.7	1.2	42.2
4 modes	17.3	20.7	22.5	19.6	19.5	26.5	18.0	21.9	17.5	22.4

Mode	**Min. objective function value**
1	0.000283
2	0.005032
3	0.010927
4	0.014635

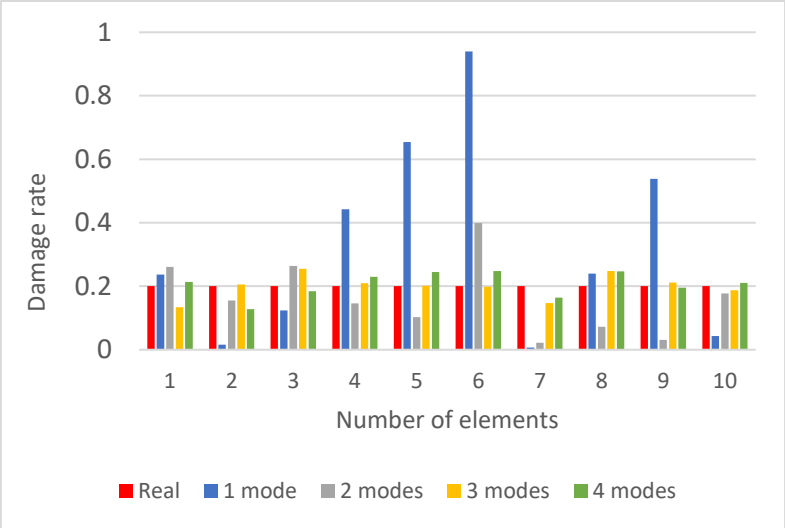

Figure 56 Damage results - Real vs Calculated (GA) using objective function y_2 with 10% noise.

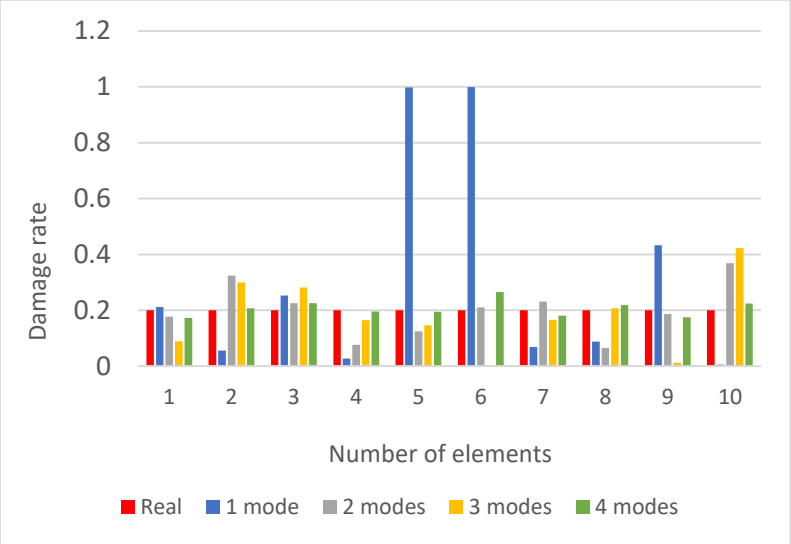

Figure 57 Damage results - Real vs Calculated (PSO) using objective function y_2 with 10% noise.

Figure 56 and Figure 57 show the performance of the second objective function (y_2) for the last damage scenario (uniform damage), when a noise of 10% is applied to the mode shapes of the structure. The objective function performs well in determining the location and scope of the damage in the uniform damage scenario. With 4 eigenmodes known, both techniques perform well overall.

e) GA Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 66 Optimization results in percentage with GA for uniform damage and 10% added noise using objective function y_3.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	54.2	40.6	47.8	33.8	47.0	95.3	19.1	59.9	72.1	0.2
2 modes	58.7	46.6	62.3	43.0	70.1	89.9	50.3	53.1	86.5	0
3 modes	47.9	52.4	44.7	45.1	57.4	79.2	32.4	48.2	73.2	0.5
4 modes	23.2	34.2	23.0	22.1	21.4	55.5	9.0	24.5	34.6	39.8

Mode	**Min. objective function value**
1	0.19012
2	1.2296
3	2.3886
4	2.4617

f) PSO Optimization with both MACFLEX and MTMAC criteria: $y_3 = \sqrt{y_1^2 + y_2^2}$

Table 67 Optimization results in percentage with PSO for uniform damage and 10% added noise using objective function y_3.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Real	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0	20.0
1 mode	36.0	33.9	38.7	0	71.9	99.9	16.0	31.0	54.8	0
2 modes	65.8	54.8	68.3	53.1	74.2	91.4	58.0	60.6	89.0	0.1
3 modes	52.7	56.4	50.5	49.6	62.7	81.6	37.8	53.2	75.5	0
4 modes	10.7	22.0	11.1	10.2	10.7	50.0	0	12.2	27.9	31.2

Mode	**Min. objective function value**
1	0.11803
2	1.1607
3	2.3522
4	2.4639

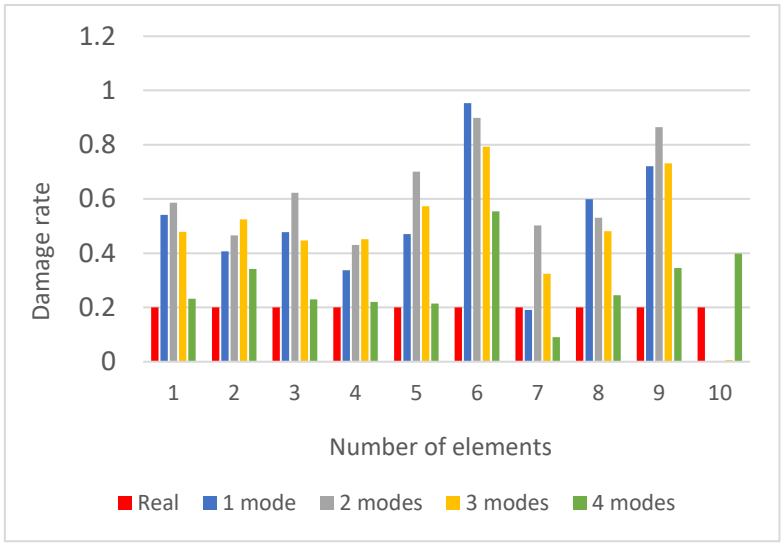

Figure 58 Damage results - Real vs Calculated (GA) using objective function y_3 with 10% noise.

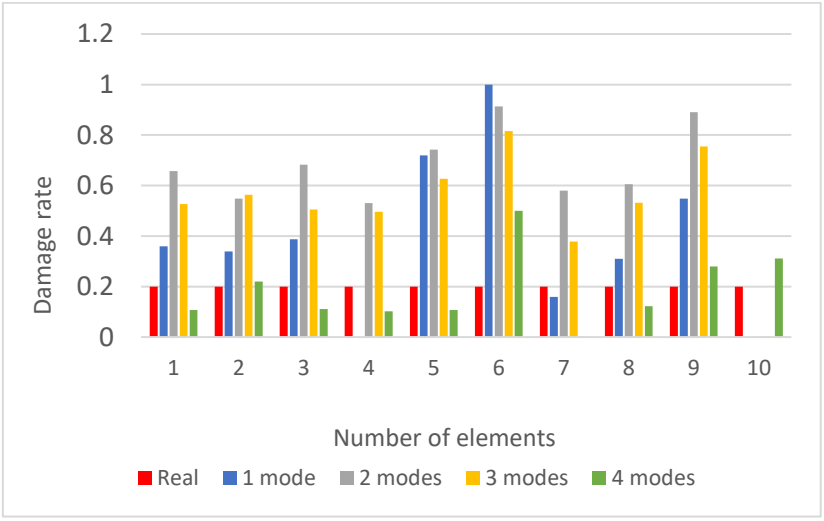

Figure 59 Damage results - Real vs Calculated (PSO) using objective function y_3 with 10% noise.

Lastly, Figure 58 and Figure 59 show the performance of the third objective function (y_3), for the last damage scenario (uniform damage), when a noise of 10% is applied to the mode shapes of the structure. The third objective function does not perform well in precisely detecting the extent or position of the damage in the structure when such high noise is introduced, similar to the other damage situations. The objective function in this situation reveals damage to all components but is unable to accurately determine its magnitude.

6.3. Discussions

The MACFLEX criteria is the foundation of the first objective function. The criteria makes use of the structure's dynamic flexibility, which may be particularly vulnerable to damage and has several benefits over stiffness. The second objective function, which is based on the MTMAC criterion, employs the natural frequencies and mode shapes of the structure in order to detect damage in structures with efficiency. The benefits of the first two objective functions are made use of by the third objective function. Using 1, 2, 3, and 4 known eigenmodes, the test applications of the present study were to determine the damage location and rate for various damage scenarios in a 10-bar planar truss structure. Additionally, The initial mode shape data is given some random noise to test the viability of each of the goal functions.

First of all, it became evident that the Particle Swarm Optimization technique is more effective than Genetic algorithm in handling these kinds of damage detection issues. Using each of the three objective functions, the PSO algorithm converged to a better solution in the test implementation of the truss structure. The algorithm is renowned for its capacity to conduct local and global searches over the solution space as well as for finding the global minimum. Additionally, PSO required substantially less time for algorithm convergence than GA did. For the evaluated objective functions, PSO thus exhibits the highest overall performance in terms of optimization outcomes and computing time.

Although it cannot provide precise damage values, the first objective function, which is based on the MACFLEX criteria, can identify the location of the damage in buildings and its extent in structural parts. When no noise was applied, the PSO was still able to provide quite accurate findings for the damage extent for the first and second damage scenarios. Due to the criteria only considering data about the eigenmodes, the optimization process fails to detect damage in the exceptional situation of uniform damage. The only attribute that changes when a structure is subjected to uniform damage is the eigenperiod, which expands, making the structure more flexible. The structure's eigenmodes remain unchanged. As a consequence, this unique kind of damage cannot be detected by the first objective function.

The second objective function, which was based on the MTMAC criteria, performed very well overall and with the PSO algorithm in particular. When 3 or 4 eigenmodes were known, this objective function was able to identify almost 100% of the location and amount of the damage for the first and second damage scenarios and about 95% of the third damage scenario (uniform damage). Because the MTMAC criteria also incorporates information about the eigenvalues (eigenperiods) and the eigenmodes of the structure, the second objective function performed much better when recognizing uniform damage in the structure. For this, the MTMAC criteria is the only one that can reliably detect this unique kind of damage. When the structure's mode shapes were subjected to high noise rates of 5% and 10%, the objective function likewise performed very well.

The third objective function, which also used the MACFLEX and MTMAC criteria, performed very well at determining the damage's location and extent in the structure. The objective function can pinpoint practically all structural damage and its degree when no noise

is present. When noise is added, the accuracy of the damage extent perceived from the findings related to the actual damage decreases as the noise level increases. The performance of the third objective can be observed to perform better than that of the second objective for all damage scenarios when no noise is applied, and vice versa when noise is applied.

7. Concluding remarks & Further work

The main goal of the present thesis was to use certain specific and readily observed dynamic characteristics to locate and quantify the damage in a 10-bar planar structure. The MACFLEX and MTMAC modal correlation criteria were used to evaluate the performance of three objective functions. The structural damage detection problem was approached as an optimization problem, and two optimization techniques were used to find solutions:

- Genetic Algorithm (GA)
- Particle Swarm Optimization (PSO)

The values of certain dynamic properties between the experimental and numerical models of the 10-bar planar structure were compared in order to minimize the objective function in this unconstrained optimization problem. Since the dynamic properties data acquired by real experiments of the experimental model was not available in the present study, the data were created by knowing the actual damage beforehand and adding noise afterwards for the simulation to better align calculated modal parameters with actual experimental values. The dynamic properties data calculated during the optimization procedure represents the numerical model.

The PSO algorithm has consistently been shown to be extremely effective and reliable, although the three objective functions displayed varying degrees of performance. PSO therefore displays the best overall performance in terms of optimization results and computation time for the analyzed objective functions.

When three or four eigenmodes were known, the second objective function, based on the MTMAC criteria, and the third objective function, based on the MACFLEX criterion, did well at pinpointing the position and extent of the damage in the structure. These objective functions nevertheless performed well when there were only one or two known eigenmodes. There were some more challenging damage scenarios than others. For instance, the first objective function failed to identify uniform damage using the PSO method, and GA was unable to provide an accurate prediction in the event of uniform damage (3rd scenario).

In summary, a simple yet quite helpful method was given to address damage detection problems. Damage, even in substantial structures, can be quickly revealed in location and extent in its structural parts, by measuring a small number of their dynamic properties. The case study has shown how reliable, quick, and simple to use the suggested technique is.

7.1. Further work

The work presented in this research study show a method that can detect a damaged position and extent in structures. The method is tested on a 10-bar planar truss structure. A suggested further work or an expansion of this study is to test the suggested method on different type of structures, such as beams, frames, and even substantial structures with large number of elements.

Another suggestion is to test the different objective functions used in this study with different optimization algorithms than the ones used here and investigate their performance. Differential Evolution (DE) and Sequential Quadratic Programming (SQP) are alternatives to investigate.

8. References

[1] A. Behtani, A. Bouazzouni, S. Khatir, S. Tiachacht, Y.-L. Zhou, and M. A. Wahab, "Damage localization and quantification of composite beam structures using residual force and optimization," *Journal of Vibroengineering*, vol. 19, no. 7, pp. 4977–4988, 2017, doi: 10.21595/jve.2017.18302.

[2] S. W. Doebling, C. R. Farrar, M. B. Prime, and D. W. Shevitz, "Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review," Los Alamos National Lab. (LANL), Los Alamos, NM (United States), LA-13070-MS, May 1996. doi: 10.2172/249299.

[3] H. Sohn, C. R. Farrar, F. M. Hemez, and J. J. Czarnecki, "A Review of Structural Health Review of Structural Health Monitoring Literature 1996-2001.," Los Alamos National Lab. (LANL), Los Alamos, NM (United States), LA-UR-02-2095, Jan. 2002. Accessed: Nov. 17, 2022. [Online]. Available: https://www.osti.gov/biblio/976152

[4] R. Hou and Y. Xia, "Review on the new development of vibration-based damage identification for civil engineering structures: 2010–2019," *Journal of Sound and Vibration*, vol. 491, p. 115741, Jan. 2021, doi: 10.1016/j.jsv.2020.115741.

[5] "Goal 9 | Department of Economic and Social Affairs." https://sdgs.un.org/goals/goal9 (accessed May 11, 2023).

[6] "Goal 13 | Department of Economic and Social Affairs." https://sdgs.un.org/goals/goal13 (accessed May 11, 2023).

[7] "Long-term renovation strategies." https://energy.ec.europa.eu/topics/energy-efficiency/energy-efficient-buildings/long-term-renovation-strategies_en (accessed May 11, 2023).

[8] I. Lawson *et al.*, "Non-Destructive Evaluation of Concrete using Ultrasonic Pulse Velocity," p. 6, 2011.

[9] A. Rytter, *Vibrational Based Inspection of Civil Engineering Structures*. in Fracture and Dynamics. Aalborg: Dept. of Building Technology and Structural Engineering, Aalborg University, 1993.

[10] "MABY5010-Group 19 (1).pdf."

[11] L. Yu and X. Chen, "Bridge Damage Identification by Combining Modal Flexibility and PSO Methods," Jan. 2010. doi: 10.13140/2.1.2129.6007.

[12] J.-C. Chen and J. A. Garba, "On-orbit damage assessment for large space structures," *AIAA Journal*, vol. 26, no. 9, pp. 1119–1126, Sep. 1988, doi: 10.2514/3.10019.

[13] S. Mukhopadhyay, H. Luş, A. L. Hong, and R. Betti, "Propagation of mode shape errors in structural identification," *Journal of Sound and Vibration*, vol. 331, no. 17, pp. 3961–3975, 2012, doi: 10.1016/j.jsv.2012.04.012.

[14] R. Allemang, "The modal assurance criterion - Twenty years of use and abuse," *Sound & vibration*, vol. 37, pp. 14–23, Aug. 2003.

[15] A. Champati, S. Voggu, and V. Lute, "Detection of Damage in Bolted Steel Structures Using Vibration Signature Analysis," *Journal of Vibration Engineering and Technologies*, 2023, doi: 10.1007/s42417-023-00916-6.

[16] M. Pastor, M. Binda, and T. Harčarik, "Modal Assurance Criterion," *Procedia Engineering*, vol. 48, pp. 543–548, Jan. 2012, doi: 10.1016/j.proeng.2012.09.551.

[17] H. P. Fuentes and M. Zehn, "Application of the Craig-Bampton model order reduction method to a composite structure: MACco, COMAC, COMAC-S and eCOMAC," *Open Engineering*, vol. 6, no. 1, pp. 185–198, 2016, doi: 10.1515/eng-2016-0024.

[18] V. Plevris and G. C. Tsiatas, "Computational Structural Engineering: Past Achievements and Future Challenges," *Frontiers in Built Environment*, vol. 4, 2018, Accessed: Apr.

16, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fbuil.2018.00021
[19] R. Adams, P. Cawley, C. Pye, and B. Stone, “A Vibration Technique for Non-Destructively Assessing the Integrity of Structures,” *Archive: Journal of Mechanical Engineering Science 1959-1982 (vols 1-23)*, vol. 20, pp. 93–100, Apr. 1978, doi: 10.1243/JMES_JOUR_1978_020_016_02.
[20] R. Perera and A. Ruiz, “A multistage FE updating procedure for damage identification in large-scale structures based on multiobjective evolutionary optimization,” *Mechanical Systems and Signal Processing*, vol. 22, no. 4, pp. 970–991, May 2008, doi: 10.1016/j.ymssp.2007.10.004.
[21] M. Georgioudakis and V. Plevris, “Investigation of the performance of various modal correlation criteria in structural damage identification,” Jun. 2016. doi: 10.7712/100016.2207.11846.
[22] D. Bernal, “Extracting flexibility matrices from state-space realizations,” Jan. 2000.
[23] H. W. Shih, D. P. Thambiratnam, and T. H. T. Chan, “Damage detection in slab-on-girder bridges using vibration characteristics,” *Structural Control and Health Monitoring*, vol. 20, no. 10, pp. 1271–1290, 2013, doi: 10.1002/stc.1535.
[24] V. Plevris, “Optimum Design of Plane Trusses Using Mathematical and Metaheuristic Algorithms on a Spreadsheet”.
[25] M. N. Suharto, M. Y. Hassan, M. S. Majid, M. P. Abdullah, and F. Hussin, “Optimal power flow solution using evolutionary computation techniques,” presented at the IEEE Region 10 Annual International Conference, Proceedings/TENCON, 2011, pp. 113–117. doi: 10.1109/TENCON.2011.6129074.
[26] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, Dec. 1995, pp. 1942–1948 vol.4. doi: 10.1109/ICNN.1995.488968.
[27] M. Younes and R. L. Kherfane, “A new hybrid method for multi-objective economic power/emission dispatch in wind energy based power system,” *International Journal of System Assurance Engineering and Management*, vol. 5, no. 4, pp. 577–590, 2014, doi: 10.1007/s13198-013-0208-z.
[28] N. P. Theodorakatos, “Fault Location Observability Using Phasor Measurement Units in a Power Network Through Deterministic and Stochastic Algorithms,” *Electric Power Components and Systems*, vol. 47, no. 3, pp. 212–229, 2019, doi: 10.1080/15325008.2019.1580801.
[29] P. R. Bergamaschi, S. F. P. Saramago, and L. S. Coelho, “Comparative study of SQP and metaheuristics for robotic manipulator design,” *Applied Numerical Mathematics*, vol. 58, no. 9, pp. 1396–1412, 2008, doi: 10.1016/j.apnum.2007.08.003.
[30] Y. He, H. Elmaraghy, and W. Elmaraghy, “A design analysis approach for improving the stability of dynamic systems with application to the design of car-trailer systems,” *JVC/Journal of Vibration and Control*, vol. 11, no. 12, pp. 1487–1509, 2005, doi: 10.1177/1077546305060832.
[31] Y. Lou, Y. Zhang, R. Huang, X. Chen, and Z. Li, “Optimization algorithms for kinematically optimal design of parallel manipulators,” *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, pp. 574–584, 2014, doi: 10.1109/TASE.2013.2259817.
[32] M. E. Ortiz-Quisbert, M. A. Duarte-Mermoud, F. Milla, R. Castro-Linares, and G. Lefranc, “Optimal fractional order adaptive controllers for AVR applications,” *Electrical Engineering*, vol. 100, no. 1, pp. 267–283, 2018, doi: 10.1007/s00202-016-0502-2.

[33] S.-B. Ma, A. Afzal, and K.-Y. Kim, “Optimization of ring cavity in a centrifugal compressor based on comparative analysis of optimization algorithms,” *Applied Thermal Engineering*, vol. 138, pp. 633–647, 2018, doi: 10.1016/j.applthermaleng.2018.04.094.
[34] A. El Mouatasim, R. Ellaia, and E. Souza de Cursi, “Stochastic perturbation of reduced gradient & GRG methods for nonconvex programming problems,” *Applied Mathematics and Computation*, vol. 226, pp. 198–211, Jan. 2014, doi: 10.1016/j.amc.2013.10.024.
[35] C. Y. P.E, “Excel Solver: Which Solving Method Should I Choose?,” *EngineerExcel*, Nov. 29, 2016. https://engineerexcel.com/excel-solver-solving-method-choose/ (accessed Apr. 20, 2023).
[36] D. Greiner, J. Periaux, D. Quagliarella, J. Magalhaes-Mendes, and B. Galván, “Evolutionary Algorithms and Metaheuristics: Applications in Engineering Design and Optimization,” *Mathematical Problems in Engineering*, vol. 2018, 2018, doi: 10.1155/2018/2793762.
[37] V. Plevris and G. Solorzano, “A Collection of 30 Multidimensional Functions for Global Optimization Benchmarking,” *Data*, vol. 7, no. 4, 2022, doi: 10.3390/data7040046.
[38] F. Udwadia, “Structural Identification and Damage Detection from Noisy Modal Data,” *Journal of Aerospace Engineering - J AEROSP ENG*, vol. 18, Jul. 2005, doi: 10.1061/(ASCE)0893-1321(2005)18:3(179).
[39] F. Sirois and F. Grilli, “Potential and limits of numerical modelling for supporting the development of HTS devices,” *Superconductor Science and Technology*, vol. 28, Dec. 2014, doi: 10.1088/0953-2048/28/4/043002.
[40] “Finite Element Analysis in MATLAB.” https://se.mathworks.com/videos/series/finite-element-analysis-in-matlab.html (accessed May 15, 2023).