

Piezoelectric Impedance-based High-accuracy Damage Identification Using Sparsity Conscious Multi-objective Optimization Inverse Analysis

Yang Zhang, Ph.D. Student

Department of Mechanical Engineering

University of Connecticut

Kai Zhou, Assistant Professor

Department of Civil and Environmental Engineering

The Hong Kong Polytechnic University

Jiong Tang[†]

Pratt & Whitney Endowed Professor

Department of Mechanical Engineering, University of Connecticut

191 Auditorium Road, Unit 3139, Storrs, CT 06269, USA

Phone: +1 (860) 486-5911; Email: jiong.tang@uconn.edu

Mechanical Systems and Signal Processing

(Accepted)

[†] Corresponding author

Piezoelectric Impedance-based High-accuracy Damage Identification Using Sparsity Conscious Multi-objective Optimization Inverse Analysis

Yang Zhang¹, Kai Zhou², Jiong Tang^{1†}

1 Department of Mechanical Engineering, University of Connecticut

2 Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University

Abstract

Two elements are essential in structural health monitoring utilizing dynamic responses: response measurement with high-frequency contents, i.e., small characteristic wavelengths, that can adequately reflect damage features, and effective inverse identification analysis that is however oftentimes under-determined. The advancement of smart structure integration has led to active interrogation through frequency-sweeping piezoelectric impedance measurement at high frequency range. In this research we develop a multi-objective optimization formulation for the identification of damage location and severity utilizing piezoelectric impedance. While one optimization objective is to match the response measurement with finite element model prediction in the damage parametric space, the other is the number of locations of damage, i.e., the sparsity of damage index as the solution vector, since damage usually occurs within a small number of locations. This multi-objective formulation fits well the under-determined nature of damage identification, as it naturally provides multiple solutions as basis for further elucidation. The challenge remaining is how to find a small solution set that can include the actual damage scenario. Here we develop a novel inverse identification framework utilizing the intelligent swarm optimizer which possesses flexibility for enhancement. We first embed a sparsity enforcement process into the population generation of the optimizer, which yields a solution repository intrinsically possessing sparsity. We then apply reinforcement learning so the agents can adaptively opt for local strategies with the aim of enriching the searching patterns to diversify the solutions. Through the incorporation of a Q-table, searching toward more promising directions will be rewarded. Our case analyses employing experimental data indicate that this sparsity-conscious multi-objective particle swarm optimization technique can lead to a small solution set which generally encompasses the true damage scenario. This effectively solves the structural damage identification problem with piezoelectric impedance measurement.

Keywords: structural damage identification, piezoelectric impedance, multi-objective optimization, sparsity, particle swarm, reinforcement learning.

[†] Corresponding author

1 Introduction

In structural health monitoring (SHM), dynamic responses of a host structure are utilized to infer the occurrence of damage and/or to identify the damage. A well-designed SHM scheme can positively impact the reliability and durability of various mechanical, aerospace, marine, and infrastructure systems. It is known that two elements are essential in SHM, the acquisition of dynamic responses carrying high-frequency contents that can be effectively extracted, and the inverse analysis of response change for the detection and identification of damage. High-frequency responses correspond to smaller characteristic wavelengths that can reflect features of small-size damage. The advent of smart materials has led to measurement advancement. The multi-field coupling of many smart transducers can facilitate near real-time active interrogation of a host structure, i.e., inducing carefully designed dynamic responses and then sensing the change of such responses as damage occurs. One type of example is the piezoelectric transducers that are compact in size, possess excellent linearity and can be easily integrated to a host structure (Park et al, 2003; Wang and Tang, 2009; Shi et al, 2022). They have been employed in a number of damage detection and identification schemes to actuate and sense global vibration, wave propagation, and electro-mechanical impedance etc (Li et al, 2019; Zhu et al, 2021; Xu and Deng, 2022). In particular, the piezoelectric impedance approach, i.e., using the same piezoelectric transducer to carry out frequency-sweeping excitation and simultaneously sense the local oscillation which then yields the piezoelectric impedance information, has shown promising aspects. The piezoelectric impedance can be measured in high-frequency range, and the harmonic responses acquired may enable model-based damage identification (Kim and Wang, 2014; Shuai et al, 2017; Cao et al, 2018; Fan et al, 2020).

Despite the progresses made in recent years, there is however still significant technological gaps in SHM especially in damage identification inverse analysis. Although in many cases the detection of damage occurrence is important, ultimately the identification of location and severity of damage can provide vital information toward operation and maintenance of the structure as well as its remaining useful life prediction. From the viewpoint of a structure being continuum, structural damage is characterized by continuous variables, exhibiting in theory infinitely many possible damage scenarios. As such, inverse analysis based on finite element modeling has an edge over classification type of approaches that typically rely on a relatively small size of damage profiles for training. Nevertheless, an intrinsic difficulty in damage identification is that the amount of useful information acquired from measurement is generally limited, oftentimes rendering the inverse analysis under-determined. In finite element model-based damage identification, a structure is divided into a large number of segments. Each segment may be subject to damage which induces local property change, e.g., stiffness reduction. The inverse analysis usually aims at solving a damage index vector. The dimension of this vector is the number of segments, and each vector component represents possible stiffness reduction percentage in a segment owing to damage occurrence.

The piezoelectric impedance is only sensitive to damage around the measurement peaks. Simply increasing the number of measurement points (i.e., frequency points) does not necessarily increase the amount of information since there is no guarantee the additional measurements are mathematically independent. Indeed, how to utilize the limited measurement to solve for the damage index vector has become a major challenge in SHM. Obviously, direct matrix inversion based on finite element model with artificial constraints such as the least square may not yield true damage scenario.

Inverse analysis can be conducted indirectly through optimization-based approaches that minimize the difference between measurement and model prediction in the damage parametric space. To tackle the issue of under-determinedness in damage identification, several investigations have resorted to imposing a physically sensible condition, i.e., the number of segments that are subject to damage is small. This is usually the case for real-world SHM especially when the damage is at its beginning stage with small-size and thus affects only a small region. Mathematically speaking the aforementioned damage index vector should be a sparse vector in which the majority of its components are zero, i.e., the damage index vector has small l_0 norm. One way of imposing this condition into a conventional single-objective optimization is through regularization with l_0 or l_1 norms of the damage index vector. The l_1 norm can be used as a convex relaxation of the l_0 norm, i.e., as a smooth approximation of the non-differentiable l_0 norm (Davenport et al, 2012). Fan et al (2018) employed the spectral projected gradient method in piezoelectric impedance sensing, where the objective function is a residual term added with l_1 regularization. Subsequently it has been recognized that population-based metaheuristic algorithms are suitable to solving such problems. For example, Ding et al (2020; 2022) utilized the Jaya algorithm to conduct damage identification, where the $l_{0.5}$ norm which is closer to the l_0 norm, was used. Similar effort using sparse representation for damage identification was carried out by Chen and Sun (2019). Although promising results have been obtained, in these single-objective optimization approaches the incorporation of the sparsity regularization involves the selection of weighting constants that are difficult to determine. It is also possible that the single solution identified may still miss the true damage scenario.

Under the umbrella of optimization-based inverse analysis, there is a different school of thought being explored, i.e., utilizing multi-objective optimization. From a computational standpoint, a multi-objective optimization naturally avoids the *ad hoc* selection of weightings between different objectives used in a single-objective optimization. Multi-objective optimization formulation has been employed in parametric identification with heterogeneous measurements for various reasons. For example, Jin et al (2014) developed a model updating formulation in which matching natural frequencies and matching mode shapes are treated as two separate objectives. To find feasible solutions under limited input information, Cha and Buyukozturk (2015) explored utilizing multiple modal strain energy (MSE) expressions as optimization

objectives to seek multiple solutions. Pereira et al (2022) investigated the optimal positioning of sensors for SHM. A bi-objective problem is analyzed that minimizes the modal response difference as well as the number of sensors. Cao et al (2018) specifically looked into incorporating sparsity of the damage index vector, the l_0 norm, as a separate objective in addition to the original objective of matching the response measurements, into damage identification.

A major difference between single objective optimization and multi-objective optimization is that the latter may yield multiple solutions. When the inverse analysis is under-determined, multiple solutions are expected mathematically. Indeed, SHM is analogous to human healthcare in many ways, including that the diagnostics practice is oftentimes under-determined. Essentially, when the measurement information is limited, one would be satisfied if the inverse analysis can produce a small set of solutions, i.e., possible damage scenarios, that encompass the true damage scenario. If all these solutions point to insignificant damage, continuing operation of the structure can be recommended. Otherwise, additional measurements using different sensing techniques including human inspection can be dispatched to further investigate this small set of possible damage scenarios. Seeking a small set of solutions that include the true damage scenario is especially relevant for piezoelectric impedance based active interrogation. On one hand, the high-frequency nature and harmonic responses lead to the possibility of model-based identification of small-size damage. On the other hand, high-frequency analysis using finite element requires high mesh density and high dimensionality of the model, leading to an even more challenging inverse analysis problem. It is worth noting that the aforementioned investigation by Cao et al (2018) appears to be the first attempt of multi-objective optimization for damage identification directly integrated with sparsity objective. In that effort, the multi-objective dividing rectangles (DIRECT) algorithm was employed, where a sigmoid transformation similar to that used in machine learning algorithms was utilized to reduce the search space. While certain level of success has been demonstrated, the DIRECT algorithm is deterministic in nature and can only handle low dimension problems, thus having limited room for tailoring toward improvement. In particular, in the context of sparsity-oriented damage identification, our goal is to find a set of possible solutions that are sufficiently diverse in order to include the true damage scenario. The quality of the solution set in terms of diversity has not been explicitly addressed in related investigations concerning the sparsity of damage index vector.

In the realm of optimization, population based intelligent algorithms such as evolutionary algorithm (Alkayem et al, 2018), simulated annealing (Yao et al, 2017), genetic algorithm (Tiachacht et al, 2018), Jaya algorithm (Ding et al, 2019), and particle swarm (Minh et al, 2021) etc have attracted considerable recent attention, as a variety of metaheuristics have been synthesized to tackle issues ranging from computational efficiency to various application-specific requirements. It is known that damage identification through inverse optimization is subject to many local extremes in the parametric space. How

to enhance searching in the optimization to escape from local extremes becomes one focal point. For example, a hybridization of the K-clustering-based Jaya algorithm and Trees Seeds algorithm was proposed to achieve robust searching (Ding et al, 2020). Similarly, it was demonstrated that the Jaya algorithm combined with Hooke–Jeeves pattern search can improve the search performance (Ding et al, 2022). Genetic Algorithm (GA) was integrated with a Modified Cornwell Indicator (MCI) and then applied to damage identification for frame structure (Tiachacht et al, 2018). Minh et al (2021) developed an enhanced particle swarm optimization, where parameters can be adjusted online to control the convergence rate in each movement of a particle and to seek global optimum. Zheng et al (2022) improved empirical learning based on the idea of motivating solution candidates to move to better positions around their own initial positions through leveraging the experience of other candidates. A known challenge in optimization-based inverse identification is that, as the number of unknowns increases, the response surface generally becomes more complicated. As such, significant research opportunities exist for evolution- or population-based optimizations designed for such applications. In particular, most of the aforementioned algorithmic advancements are based on single-objective formulation. Enhancing the solution quality in multi-objective optimization tailored for damage identification application remains to be explored systematically.

Meanwhile, enabling optimization solution with sparsity has been explored in various applications, including feature selection, sparse regression, critical node detection in networking, and neural network training etc. Feature selection aims to identify a subset of relevant features for a given target concept (Dash and Liu, 1997). Xue et al (2012) extended the particle swarm optimization by incorporating nondominated sorting, mutation, crowding distance, and dominance to solve a bi-objective feature selection problem. Sparse regression seeks a sparse approximation solution to the regression problem, with the solution vector containing only a few non-zero elements. Qian et al (2015) proposed a Pareto optimization for subset selection that simultaneously optimizes mean square error and subset size. Ventresca et al (2018) formulated a bi-objective critical node detection problem in networking to maximize the number of connected components and minimize the variance of their cardinalities. Three well-known multi-objective indicators, i.e., additive epsilon, hypervolume, and spacing, are used to measure the quality, and spread of solutions on the approximation front. They demonstrated that the NSGA-II algorithm may outperform other approaches on benchmark problems. Tian et al (2019) developed a sparse evolutionary algorithm to optimize the weight parameters involved in neural network training. While their method inherits the fundamental ideas from the NSGA-II algorithm, they specifically designed techniques to initialize the population and generate offspring to ensure sparsity.

Building upon previous investigations, in this research we aim at the establishment of a new sparsity conscious multi-objective optimization framework that can fully unleash the capability of piezoelectric impedance measurement for damage identification. We choose the multi-objective particle swarm

optimization (MOPSO) as the basic approach, owing to its fast convergence and in particular the capability of obtaining a set of trade-off solutions in a single run which offers the potential for designing local search strategies to enrich the search patterns. We start from a sparse population generation technique that can yield the initial sparsity of the damage index vector and then iterate during the solution procedure. This can tackle the issue of high dimensionality in inverse identification. Moreover, we incorporate local searches with a time-varying inertia weight and two sets of acceleration coefficients. Both sets form two distinct coefficient-based local search strategies that can facilitate firstly exploration and subsequently exploitation. Additionally, we develop three mutation-based local search strategies to trigger a local particle movement to encourage exploration abilities, resulting in a diverse set of solutions. Indeed, imposing sparsity matches with the damage identification nature, and it can also impact the search process toward the optimal solutions. As such, it provides the opportunity to ensure that the parametric space is searched thoroughly to yield a small but diverse solution set. Here we integrate Q-Learning, which is a reinforcement learning policy that will find the next best action, given a current state, into MOPSO. It chooses the action based on the maximum Q-table value and aims to maximize the reward for the achievement, i.e., finding a better solution. With this reinforcement learning, each particle is considered as an agent and each agent will have multiple search methods (local searching strategies). Through Q-table, the agents can interact well with the search environment as well as other agents, thus enhancing the global search capability. All the strategies are not executed simultaneously. Instead, a promising one will be selected at the current iteration according to the Q-learning algorithm. The combination of sparse population generation and reinforcement learning based strategies for solution exploration will yield a small solution set that includes the true damage scenario.

The rest of the paper is organized as follows. In Section 2, finite element modeling of piezoelectric impedance for structural health monitoring is outlined, followed by the problem formulation of damage identification emphasizing sparsity of damage index vector. Section 3 elaborates the multi-objective optimization solution procedure, which includes the sparse population generation and iteration, and the integration details of Q-learning with MOPSO. In Section 4, an experimental case utilizing the new algorithm is presented. The case setup with data acquisition from the physical testbed and the corresponding finite element analysis are described in detail, followed by inverse identification of damage using experimental data. The solution quality is demonstrated. Section 5 gives the concluding remarks.

2 Piezoelectric Impedance-based Damage Identification

In this section, we present the formulation of damage identification utilizing piezoelectric impedance measurement. As the identification is model-based, we first outline the finite element modeling of piezoelectric impedance prediction/simulation. We then formulate a multi-objective optimization problem that will be used to identify the location and severity of damage in a structure.

2.1 Finite element modeling of piezoelectric impedance

When a piezoelectric transducer is integrated to a host structure, the coupled structure-transducer interaction is characterized by the following equations (Shuai et al, 2017; Cao et al, 2018)

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} + \mathbf{K}_{12}Q = \mathbf{0} \quad (1a)$$

$$R\dot{Q} + k_c Q + \mathbf{K}_{12}^T \mathbf{q} = V_{in} \quad (1b)$$

where $\mathbf{q}$ is the displacement vector, k_c is the inverse of the capacitance of the piezoelectric transducer; $\mathbf{K}_{12}$ is the electromechanical coupling vector between the transducer and the structure, and $\mathbf{K}$, $\mathbf{C}$, and $\mathbf{M}$ are the stiffness, damping, and mass matrices, respectively. In this research, we use piezoelectric admittance which is the reciprocal of impedance as information carrier for damage identification. Specifically, we apply frequency-sweeping harmonic voltage excitation, denoted as V_{in} , to the transducer. Q is the electrical charge on the surface of the piezoelectric transducer, and R is the resistance in the measurement circuitry. It is worth noting that in the experimental implementation, we measure the voltage drop across the resistor, which is denoted as V_{out} , and subsequently obtain the current as $\dot{Q} = V_{out} / R$. As such, here we include an additional term $R\dot{Q}$ in Equation (1b), compared with those presented in previous research (Shuai et al, 2017; Cao et al, 2018). We let the excitation frequency be denoted as ω , and use overbar hereafter to indicate magnitude of the corresponding response variable. The piezoelectric admittance of the healthy structure, which is the ratio of the current magnitude and the voltage magnitude, can be derived as

$$y_h(\omega) = \frac{\dot{\bar{Q}}}{\bar{V}_{in}} = \frac{j\omega\bar{Q}}{\bar{V}_{in}} = \frac{j\omega}{j\omega R + k_c - \mathbf{K}_{12}^T (\mathbf{K} + j\omega\mathbf{C} - \omega^2\mathbf{M})^{-1} \mathbf{K}_{12}} \quad (2)$$

where j is the imaginary unit.

In a finite element model for damage identification, the structure is generally divided into n segments to facilitate the identification of location and severity of damage. That is, each segment is susceptible of damage occurrence, and damage causes homogenized change of structural properties within a segment. In this research, without loss of generality, we assume damage causes stiffness reduction in one or multiple segments. As such, the stiffness matrix with structural damage occurrence, $\mathbf{K}_d$, can then be expressed as

$\mathbf{K}_d = \sum_{i=1}^n \mathbf{K}_h^i (1 - \alpha_i)$, in which $\mathbf{K}_h^i$ is stiffness matrix of the i -th segment under the healthy status and the

summation refers to the direct summation in finite element model assemblage. $\alpha_i \in [0,1]$ is the damage index indicating the possible stiffness loss of the i -th segment, which is to be identified. The piezoelectric admittance corresponding to the damaged structure can be written as

$$y_d(\omega) = \frac{j\omega}{j\omega R + k_c - \mathbf{K}_{12}^T (\mathbf{K}_d + j\omega \mathbf{C} - \omega^2 \mathbf{M})^{-1} \mathbf{K}_{12}} \quad (3)$$

Mathematically, the change of admittance can be derived based on Equations (2) and (3). To increase the computational efficiency, based on the assumption of damage being small in size, we apply the Taylor series expansion to the expression of the change of admittance and ignore the higher-order terms, i.e.,

$$\Delta y(\omega) = \sum_{i=1}^n [j\omega(j\omega R + k_c - \mathbf{K}_{12}^T \mathbf{Z}^{-1} \mathbf{K}_{12})^{-2} \mathbf{K}_{12}^T \mathbf{Z}^{-1} (\mathbf{L}_i^T \mathbf{K}_{el} \mathbf{L}_i) \mathbf{Z}^{-1} \mathbf{K}_{12}] \alpha_i \quad (4)$$

where $\mathbf{Z} = \mathbf{K} - \mathbf{M}\omega^2 + j\omega \mathbf{C}$ represents the dynamic stiffness of the structure, and $\mathbf{L}$ is the Boolean matrix indicating how the segmental stiffness matrices are assembled into the global stiffness matrix. Once again, the term $j\omega R$ corresponds to the measurement resistance employed in experiment. This formulation is completely consistent with the actual experimental data acquisition procedure. In Equation (4), the change of admittance under excitation frequency ω is denoted as $\Delta y(\omega)$. In damage identification practice, we conduct frequency sweeping in the frequency range of interest, and acquire a series of admittance changes $\Delta y(\omega_1), \dots, \Delta y(\omega_m)$, under m different excitation frequencies $\omega_1, \dots, \omega_m$. We introduce the following notations of excitation frequency vector, admittance change vector, and damage index vector,

$$\boldsymbol{\omega} = [\omega_1, \dots, \omega_m]^T \quad (5a)$$

$$\Delta \mathbf{y}(\boldsymbol{\omega}) = [\Delta y(\omega_1), \dots, \Delta y(\omega_m)]^T \quad (5b)$$

$$\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_n]^T \quad (5c)$$

Based on Equation (4), we can then obtain the following relation,

$$\Delta \mathbf{y} = \begin{bmatrix} \Delta y(\omega_1) \\ \vdots \\ \Delta y(\omega_m) \end{bmatrix} = \mathbf{T}_{m \times n} \boldsymbol{\alpha}_{n \times 1} \quad (6)$$

where $\mathbf{T}$ is the finite element-based sensitivity matrix that links the admittance change vector with the damage index vector.

2.2 Formulation of sparsity conscious multi-objective optimization for damage identification

In actual practice of damage identification, we conduct model updating so the finite element model of the healthy structure can match the actual structure. We then conduct piezoelectric admittance measurement to obtain the admittance change vector $\Delta \mathbf{y}$ with respect to the healthy admittance. However, it is worth emphasizing that, while the damage index vector $\boldsymbol{\alpha}$ and the admittance change vector is related directly as shown in Equation (6), this equation for solving for the damage index vector is oftentimes under-determined. In the admittance versus frequency curve, the peaks correspond to structural resonances. Structural damage

manifests itself only in the vicinity of the resonant peaks in the admittance measurements, which means the information for damage inference is generally limited. Although one can increase the number of sweep frequency points of admittance measurement, there is no guarantee that such an increase can lead to additional information. In other words, simply increasing the number of measurement points does not necessarily increase the row rank of the sensitivity matrix $\mathbf{T}$. This is especially true for admittance measurements at high frequency range. In order to achieve high-precision identification of small size damage, the finite element mesh density is high and the segment size is usually chosen to be relatively small, leading to large number of segments, n .

Aiming at tackling this issue, in this research we cast the inverse identification into a multi-objective optimization formulation. One objective is obvious, i.e., we need to minimize the difference between the measured admittance change, $\Delta\mathbf{y}$, and the model prediction in damage parametric space, $\mathbf{T}\boldsymbol{\alpha}$. In practical situation, damage usually occurs within a small region of the structure, especially when the damage is at its beginning stage with small size. Therefore, we introduce another objective function that the damage index vector $\boldsymbol{\alpha}$ is sparse with small l_0 norm. The optimization problem then takes the following formulation with two objective functions,

$$\text{Find: } \boldsymbol{\alpha} \in \mathbf{E}^n, \alpha_i \leq \alpha_i \leq \alpha_u, i = 1, \dots, n \quad (7a)$$

$$\text{Minimize: } f_1 = \|\Delta\mathbf{y} - \mathbf{T}\boldsymbol{\alpha}\|_2 \text{ and } f_2 = \|\boldsymbol{\alpha}\|_0 \quad (7b)$$

where $\|\cdot\|_2$ and $\|\cdot\|_0$ denote, respectively, the l_2 norm and the l_0 norm. It is worth noting that conventional regularization methods for enforcing the sparsity in single-objective optimization use for example the l_2 and the l_1 norms. However, the former may spread out the error almost evenly among all variables and return a non-sparse vector. Researchers often opt for the l_1 norm regularization as a compromise between sparsity and computational efficiency, but the l_1 norm is only a relaxation approximation of the l_0 norm under certain conditions (Davenport et al, 2012; Cao et al, 2018; Fan et al, 2018). The conditions required may not always hold. In contrast, in this research we use the l_0 norm to achieve true sparsity since the l_0 norm directly penalizes the number of non-zero components. As we cast the problem into the above multi-objective optimization, we avoid the difficulty in selecting weightings needed for single-objective optimization, and as illustrated in the subsequent section we can indeed solve for sparse damage index vector directly though minimizing the l_0 norm. Another major advantage of casting the inverse identification into a multi-objective optimization is that such a formulation naturally yields multiple solutions, which fits the nature of SHM. These multiple solutions correspond to possible damage scenarios. If all the results point to insignificant severity, continuing operation can be expected. If certain damage

scenarios identified within the optimal solution set raise concern, we may either adopt proper maintenance decision or dispatch additional sensor measurements and human inspection.

2.3 Multi-objective particle swarm optimization (MOPSO) procedure and challenges

Our goal is to develop an effective algorithm to solve the multi-objective optimization problem formulated as Equations (7a) and (7b). In recent years, significant attention has been paid to integrating various metaheuristics into population based intelligent optimization algorithms tailored for various applications. Here we specifically choose the MOPSO as the basic approach, owing to its fast convergence and in particular, as will be shown in the subsequent sections, the capability of obtaining a set of trade-off solutions in a single run. This offers the opportunity to significantly improve the solution quality and diversity for damage identification inverse analysis with sparsity. Particle swarm is a stochastic technique originally developed by Eberhard and Kennedy (1995). It simulates the social behavior of animals such as insects, herds, birds, and fishes. These swarms conform a cooperative way, and each member in the swarm keeps changing the search pattern according to the learning experiences of its own and other members.

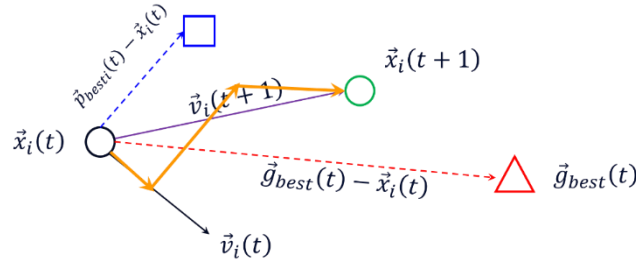


Figure 1. Illustration of particle swarm.

The working mechanism of particle swarm optimization is illustrated in Figure 1, in which each particle will fly toward the personal best in its memory and to the global best among all the particles. Thus, the new velocity and position can be expressed mathematically as

$$v_i(t+1) = zv_i(t) + c_1 r_1 (p_{best_i}(t) - x_i(t)) + c_2 r_2 (g_{best}(t) - x_i(t)) \quad (8)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (9)$$

where v is the velocity of the particle, x is the position of particle or the solution of the optimization problem, r_1 and r_2 are random numbers, t is the current iteration, p_{best} is the personal best, and g_{best} is the global best. In the above equations, coefficient z is the inertia weight, c_1 is the coefficient of the cognitive component which indicates that each particle learns from its experience, and c_2 is the coefficient of the social component from which all particles learn. While having been used in various applications,

conventional particle swarm optimization may suffer from premature convergence and low exploration ability (Gad, 2022). The inverse analysis formulated as optimization problem shown in Equations (7a) and (7b) is challenging as the damage index vector generally has high dimension. How to embed sparsity into the optimization procedure and how to escape from local minima in such a problem are particularly challenging to deal with.

3 Solving Sparsity Conscious Multi-objective Particle Swarm Optimization

Particle swarm optimization employs a single search strategy in the search process to quickly converge to an optimal local region. During iterative calculation, particles tend to gather around the local extremes. In the absence of external interference, particles may become trapped in a valley with poor solution quality. On the other hand, while reinforcement learning allows for the selection of desired actions, the absence of a fitness function in these methods makes it difficult to determine how quickly the agent (i.e., particle) can control the system's state to achieve a set of goal states (Meerza et al, 2019). To address this, in this research we combine the features of reinforcement learning and the MOPSO process. This combination allows for action selection by incorporating the Q-Table into the fitness assessment of the MOPSO. As a result, agents can find the optimal path and learn the environment more efficiently with less complexity. Throughout the solution process, we embed a sparse population generation technique into the MOPSO, which enables the initialization and guidance of the solution sparsity of the damage index vector.

3.1 Overall architecture of MOPSO guided by Q-learning

In damage identification, the damage usually occurs in only a small number of segments, which corresponds to a sparse damage index vector. To improve upon the conventional particle swarm optimization, we start from a sparse population generation technique that can yield the initial sparsity of the damage index vector and then iterate during the solution procedure. Meanwhile, conventional MOPSO neglects the variability among particles of different generations and applies uniform and fixed parameters. Here we propose five local search strategies, aiming at emphasizing the exploration and exploitation capabilities through coefficient variations and increasing the solution diversity through mutations. To adaptively select appropriate strategies, we integrate Q-learning (Watkins and Dayan 1992) into MOPSO. Each particle selects the most suitable strategy for its own iteration and evolution based on the value of the Q-table. The entire optimization procedure interacting with Q-learning is shown in pseudo-code Algorithm 1. Motivated by the memetic algorithm (Samma et al, 2016), here we formulate a procedure to deal with multi-objective optimization problems with sparsity features. As detailed in Algorithm 1, after initializing the population using Algorithm 2 (to be elaborated in Section 3.2), the personal best, the global best, and the Q-table are initialized with all zeros for the entries corresponding to the states and their possible actions. The details of the Q-learning and Q-table will be given in Section 3.3. The current state is set as *Exploration*

because the particle is expected to explore more searching space at the beginning of the algorithm. In the main iteration part, the particle (agent) will select the action which has the maximum value in the Q-table and execute the action. In a single optimization problem, the immediate reward can be obtained after executing the action if the condition $f(x_{new}) < f(x_{old})$ is satisfied. However, in multi-objective optimization, observing the fitness improvement of objectives one by one is not enough to get the reward since the objectives may conflict with each other. Therefore, the dominating algorithm is used to determine if the reward or a penalty is given after the action is executed. If the new solution dominates the old solution, then the agents receive a reward; otherwise the agents get a penalty. The reward or penalty for executing an action is given in Section 3.4.

Algorithm 1: Framework of Proposed Sparsity Conscious MOPSO

```

1  Randomly initialize each population using Algorithm 2
2  Initialize  $p_{best}$  and  $g_{best}$ 
3  Initialize Q table using Algorithm 3
4  Initialize current state  $s_t$  as  $s_{Exploration}$ 
5  for  $ii = 1$  to  $MaxFEs$  do
6      for  $jj = 1$  to  $N$  do
7          Select the best action for current state from Q table
8          Enter action repository and execute corresponding action using Algorithm 4
9          Obtain the immediate reward after executing the action
10         Observe the new state  $s' = \delta(a, s)$ 
11         Update the Q table entry and current state using Algorithm 3
12         Update external optimal solution repository
13     End
14 End

```

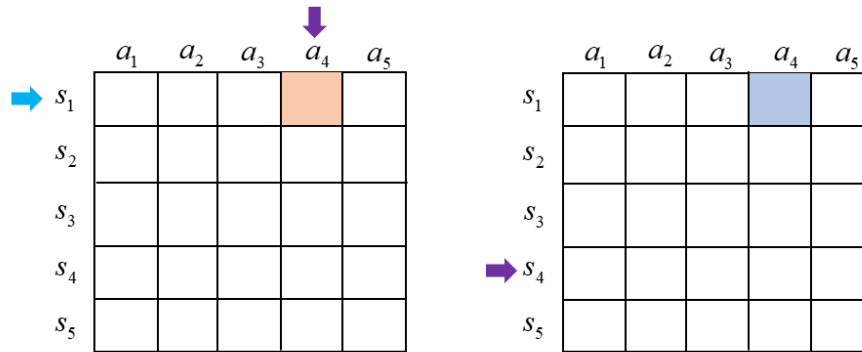


Figure 2. Illustration of the state transition process.

After the current action is executed, the agent can reach the new state using the observing function $\delta(s, a)$ which makes a transition from the current state to another. The transition here is that the index of the current action is the index of the next state. A diagram is drawn to show how to switch from the current state to the next state. As shown in Figure 2, the agent will select an action based on the maximum Q-table values given the current state as S_1 . Let's assume that action 4 (a_4) has a maximum value and then the next state will be set as S_4 , which has the same index as the index of the current action a_4 . By iterating this process, a dynamic selection and execution can be achieved between these states and actions.

After action 4 is executed, the rewards currently obtained can be used to update the entry for the current state and current action. Details about entry updating using rewards obtained will be given in Section 3.3. Then, the personal and global best are updated, and the external optimal repository will be updated accordingly. All the steps in the main loop will be performed until the maximum number of runs are reached.

3.2 Sparse population generation

In intelligent optimization, an initial population is created at the beginning as a set of candidate solutions. The population comprises individuals, each representing a possible solution. In our problem of damage identification, damage usually occurs in a small number of segments and the damage index vector expressed as Equation (5c) is thus sparse. If a specific component of the damage index vector is greater than 0, it indicates that the corresponding segment (i.e., the location) has non-zero stiffness reduction. It is worth noting that the occurrence of damage in a segment is a binary concept, described as either 0, no damage, or 1, with damage. We can utilize this to create a sparse population generation procedure so that not all unknowns will have non-zero values.

In this sub-algorithm of sparse population generation, we temporarily split the damage index information into two parts: (1) damage location, expressed as a damage location index and denoted as $\theta_1 \in \{1, \dots, n\}$; and (2) damage severity level, expressed as a percentage change in stiffness and denoted as $\theta_2 \in [0, 1]^T$. By using the element-wise product, we can then have $\alpha = \theta_1 \odot \theta_2$. Here θ_1 is a binary vector with all elements being as 0 or 1 to indicate if there is damage or not at the corresponding segment, and θ_2 is a regular vector with continuous values to indicate the possible damage severity. Tian et al (2019) proposed an evolutionary algorithm to deal with optimization in feature extraction and neural network training. In that research, a sparse population is generated during the initialization. This technique is adapted as shown in pseudo-code Algorithm 2 and implemented in our optimization framework.

Algorithm 2 mainly includes two parts: lines 1-6 acquire the rank (to be explained subsequently) using the non-dominated sorting (Deb et al 2000) for each damage location, and the remaining part generates a

sparse population based on the rank obtained. In the first part, we randomly initialize a temporary full-rank severity matrix $\mathbf{S}$, the size of which is n by n , with n as number of segments. Then we generate an n by n identity matrix for damage occurrence $\mathbf{B}$ and obtain a temporary population using the element-wise product. One feature of this temporary population is that only the elements on the diagonal have values. The purpose of this is to let each damage location make contribution to the fitness function independently. The fitness values are then passed into the Non-Dominated Sorting algorithm to return the ranks. So now each damage location gets one attribute called rank. In the remaining part, we re-initialize the severity matrix $\mathbf{S}$ with size of N by n . Here N is the number of populations. Different from the previous part, all entries of matrix $\mathbf{B}$ with size of N by n are set as zeros at this point. Then, two damage locations randomly selected are compared with each other using the ranks obtained in the previous part. The damage location with a smaller rank will be set as 1 in its $\mathbf{B}$ matrix component since smaller rank means the solution is closer to true damage scenario. By doing this for all the populations, we obtain a sparse population matrix with size of N by n using element-wise product. This returned population not only indicates the damage locations and severities but also has sparsity feature. This population will be passed to the main iteration loop for evolution.

Algorithm 2: Sparsity Population Generation

Input: number of population N , number of variables n , lower bound of variable $vMin$, upper bound of variable $vMax$
Output: Population

- 1 Initialize a matrix randomly for severity $\mathbf{S} = rand_{n \times n}$
- 2 Initialize an identity matrix for occurrence of damage $\mathbf{B} = eye_{n \times n}$
- 3 **for** $ii = 1$ to $length(\mathbf{S})$ **do**
- 4 temp.Pop = $\mathbf{S} \odot \mathbf{B}$
- 5 temp.Cost = $CostFunction(temp.Pop)$
- 6 **End**
- 7 Rank number $F \leftarrow NonDominatedSorting(temp)$
- 8 Initialize variable matrix as $\mathbf{S} = (vMax - vMin) \times unifrnd(0, 1, [N, n])$
- 9 Initialize a binary matrix as $\mathbf{B} = zeros(N, n)$
- 10 **For** $jj = 1$ to N **do**
- 11 $Ta = randperm(n, 2)$
- 12 $a = Ta(1)$
- 13 $b = Ta(2)$
- 14 **If** $F_a < F_b$ **then**
- 15 $\mathbf{B}(jj, a) \leftarrow 1$
- 16 **else**
- 17 $\mathbf{B}(jj, b) \leftarrow 1$
- 18 **End**
- 19 **End**
- 20 Population = $\mathbf{S} \odot \mathbf{B}$
- 21 **Return** Population

3.3 Q learning details and Q-table

In Algorithm 1, which is the main algorithm, Q-learning is utilized to guide particles to adaptively choose the suitable local search strategy for its own iteration. Here in this sub-section, we elaborate on how Q-learning works and how particles interact with the environment using the Q-table. Q-learning falls under the class of reinforcement learning techniques (Watkins and Dayan, 1992). In reinforcement learning, the learner performs an action causing a state transition in the environment in which it resides and receives a reward or penalty for the action executed to reach a definite goal. The environment is the searching space of the particles. The states are current local searches. Two sets of acceleration coefficients, one prioritizing self-learning and the other prioritizing social-learning, are used with a time-varying inertia weight. These form distinct coefficient-based local search strategies that can facilitate firstly the exploration and subsequently the exploitation. Furthermore, three mutation-based local search strategies are formulated to encourage local particle movement. The first two randomly perturb a single dimension of either the personal best or the global best solution. The last strategy tunes each dimension of the personal best through mutation. To be concise, we name these five local search methods as *Exploration*, *Convergence*, *Elitist-based perturbation (EBP)*, *Mutation*, and *Fine Tuning*. An action is the transition from one state to another. In the integrated algorithm, each particle of MOPSO is regarded as an agent. Updating of the Q-table occurs after each action is executed and ends when an episode is completed. The agent will not learn much after a single episode, but eventually, with sufficient exploring steps and episodes it will converge and learn the optimal Q values. The pseudo-code of the Q-learning is given in Algorithm 3.

Algorithm 3: Q-learning

```

1  For each state  $s$  and action,  $a$ 
2      Initialize  $Q(s, a) \leftarrow 0$ 
3      Initialize current state  $s_t$ 
4  End
5  Repeat
6      Select the best action  $a_t$  for  $s_t$  from the Q table and execute it
7      Receive an immediate reward  $r(s, a)$ 
8      Observe the new state  $s' = \delta(a, s)$ 
9      Update the table entry using Equation (11)
10     Update current state  $s = s'$ 
11 Until  $MaxFEs$  is reached

```

There are 5 states and their corresponding actions. Therefore, we can create a Q-table with the size of 5 by 5 and initialize all the entries as zeros. We then reset all the values corresponding to different states in action *Fine Tuning* as -5, ensuring a higher priority for other actions to be executed. Usually, the current

state is randomly chosen from the state repository. However, here we initialize the current state as *Exploration* since we demand the algorithm to start with exploring more searching space. Subsequently, the best action will be selected and executed based on the maximum Q-table value given the current state. This process can be written mathematically as

$$a_t = \text{index}\{\max(Q(s_t, \text{actions}))\} \quad (10)$$

Here, a_t is the current action and $t = 1, \dots, 5$. After the action selected is executed, an immediate reward or penalty will be obtained. Reward or penalty is determined by the dominating algorithm, which will be discussed in the following sub-section. The agents observe the maximum future reward under the next state. The next state index is the same as the current action index, as depicted in Figure 2. The reward/penalty obtained will be passed into the updating function to update the Q-table entry $Q(s_t, a_t)$. The updating function is as

$$Q_{\text{new}}(s_t, a_t) = (1-b)Q(s_t, a_t) + b \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a_{\text{all}}) \right] \quad (11)$$

where b is called the learning rate which is defined as how much the new value is accepted, γ is a discount factor used to balance immediate and future rewards, and r is the reward received for taking action a_t at a given state s_t . In Equation (11), max is the operation of taking the maximum of the future reward from the new state s_{t+1} . Finally, the current state will be updated. By iterating this process, the optimal Q-table can be achieved.

3.4 Action repository

From the perspective of structural damage identification, the objective is to acquire a small but diverse solution set that covers the true damage scenarios. In addition to the sparse population generation algorithm presented in Section 3.2, the algorithms pursuing robust exploration capabilities are also sought. In this sub-section we present five local search strategies, including two coefficient variation-based strategies and three mutation-based strategies, which collectively aim at encouraging particle movement around local extremes and facilitating exploration for a diverse set of solutions. These strategies form an action repository so that each particle can select a proper one in each iteration. The pseudo-code for the action repository is shown in Algorithm 4.

Algorithm 4: Action Selection

- 1 **Switch** current action
 - 2 **Case:** ‘Exploration’
 - 3 | Update position and velocity of particle using Equations (8) and (9)
-

4 Deal with bounds
5 Update p_{best} if *Dominates* ($x_{new}, x_{current}$) is true; otherwise, do nothing
6 $reward \leftarrow -1$ if *Dominates* ($x_{new}, x_{current}$) is true; otherwise, $reward \leftarrow -2$
7 **Case: ‘Convergence’**
8 Implement all steps same to case ‘Exploration’ only with different coefficients
9 **Case: ‘EBP’**
10 Update position of personal best using equation in Table 1
11 Deal with bounds
12 Update p_{best} if *Dominates* ($x_{new}, x_{current}$) is true; otherwise, do nothing
13 $reward \leftarrow -1$ if *Dominates* ($x_{new}, x_{current}$) is true; otherwise, $reward \leftarrow -2$
14 **Case: ‘Mutation’**
15 Mutate position of global best using equation in Table 1
16 Deal with bounds
17 Update g_{best} if *Dominates* ($x_{new}, x_{current}$) is true; otherwise, do nothing
18 $reward \leftarrow -1$ if *Dominates* ($x_{new}, x_{current}$) is true; otherwise, $reward \leftarrow -2$
19 **Case: ‘Fine Tuning’**
20 Update p_{best} using Equations (13)-(15)
21 $reward \leftarrow -3$
22 **End**

Table 1. Mathematical expressions of the actions.

Strategy Name	Particle dynamics	Coefficients
Exploration	$v_i(t+1) = \omega v_i(t) + c_1 r_1 (p_{best_i}(t) - x_i(t))$ $+ c_2 r_2 (g(t) - x_i(t))$ $x_i(t+1) = x_i(t) + v_i(t+1)$	$\omega = \omega_{max} - \frac{(\omega_{max} - \omega_{min})it}{maxIter} \quad c_1 = 2.5$ $c_2 = 0.5$
Convergence	$v_i(t+1) = \omega v_i(t) + c_1 r_1 (p_{best_i}(t) - x_i(t))$ $+ c_2 r_2 (g(t) - x_i(t))$ $x_i(t+1) = x_i(t) + v_i(t+1)$	$\omega = \omega_{max} - \frac{(\omega_{max} - \omega_{min})it}{maxIter} \quad c_1 = 0.5$ $c_2 = 2.5$
Elitist-based perturbation	$tempP_{Best_i,d} = \begin{cases} P_{Best_i,d} + r_d \cdot (vMax_d - vMin_d) & \text{if } rand > 0.5 \\ P_{Best_i,d} - r_d \cdot (vMax_d - vMin_d) & \text{if } rand \leq 0.5 \end{cases}$ repeated for defined times	$N \sim (\mu, R^2)$ $R = R_{max} - \frac{(R_{max} - R_{min}) \cdot it}{maxIter}$
Mutation	$tempL_{Best,d} = L_{Best,d} + normrnd(\mu, R^2)$ repeated for defined times	$N \sim (\mu, R^2)$ $R = R_{max} - \frac{(R_{max} - R_{min}) \cdot it}{maxIter}$
Fine Tuning	$tempP_{Best_i,d} = P_{Best_i,d} + V_{i,d}$ repeated for all dimensions	—

The actions involved are described mathematically and summarized in Table 1. For the *Exploration*, we hope that the particle can explore more searching space and avoid premature convergence. Therefore, the coefficients in Equation (8) are set as $c_1 > c_2$. In the *Convergence* state, the algorithm will converge to the global best quickly. Therefore, the social part now is given a larger value, so $c_1 < c_2$. Since the inertial weight z controls the balance between the global and local searches, it is set as a decay expression to search more space initially and at the end of the optimization will realize quick convergence. Here $z_{\max} = 0.9$ and $z_{\min} = 0.4$ are recommended (Kennedy and Eberhart, 1995). Whenever the action is selected and executed, the newly generated solution will be compared to the old one using the non-dominated sorting algorithm, and the immediate reward will be obtained as shown below

$$\text{if } x_{\text{new}} \prec x_{\text{old}}, x_{\text{old}} = x_{\text{new}} \quad \text{reward} \quad (12a)$$

$$\text{if } x_{\text{new}} \sim\prec x_{\text{old}}, x_{\text{old}} = x_{\text{old}} \quad \text{penalty} \quad (12b)$$

If the newly obtained solution dominates the old solution, a reward will be assigned to the corresponding action and the old solution will be replaced by the new one. Otherwise, the old solution remains, and a penalty will be given for the action selected. Here the reward is set as a negative number. Since we initialize the Q-table with all zero entries, if in the beginning the *Exploration* receives a reward, e.g., reward = 1, then the algorithm will be trapped there because we use the max Q-table value to select the action.

To alleviate the premature convergence issue, an elitist-based perturbation (*EBP*) strategy is adopted to perform perturbation on the personal best of the particle and provide extra diversity for it to jump out from the local optima (Lim and Isa, 2016). Here we establish a variable, *count*, to facilitate the repetition of the *EBP* strategy for a designated number of iterations, thereby enabling the activation of *EBP* when required. In this research, *count* = 10. In the *EBP* strategy, a randomly selected dimension in the personal best of the particle is perturbed using the expressions provided in Table 1. Here $P_{\text{Best},d}$ means the d -th dimension of the personal best of the i -th population. r_d is a random number obtained from the normal distribution of $N \sim (\mu, R^2)$ with $\mu = 0$ and standard deviation R . Here, $R_{\max} = 1$ and $R_{\min} = 0.1$ as suggested in literature (Lim and Isa, 2016) are the maximum and minimum perturbation ranges, respectively. it is the current number of iterations and maxIter is the maximum iteration of the algorithm. Still, the personal best will be replaced by a temporary personal best if the latter dominates the former and the immediate reward will be given as -1; otherwise, the personal best remains unchanged, and the reward will be -2 as a punishment.

In the action *Mutation*, emphasis is placed on the global best, facilitating local exploration and the possibility of identifying and transitioning to a superior position. Since the global best guiding all the particles in the MOPSO algorithm is like a leader, we use L to represent the global best. As shown in Table 1, $L_{\text{Best},d}$ indicates the d -th dimension of the global best. A random perturbation is generated based on the

same norm distribution in strategy *EBP*. The decay standard deviation is used to achieve a high jump at the beginning and a low jump at the end of the algorithm. At this time, if the temporary global best dominates the current global best, it will replace the current one. The immediate reward is now given as 1. Since the global best plays a crucial role in guiding all the particles, a large reward is given to it when it gets improvement. Otherwise, the current global best remains the same and a penalty is given as -2.

In the *Fine Tuning* process, the action focuses on each dimension of the personal best instead of only one randomly selected dimension by updating its velocity. In other words, we expect the particles to move more in the direction favorable to improve the solution and, conversely, to move more slowly in the direction unfavorable to improve the solution. For each dimension, the tuning will be repeated h times to trigger the action. h cannot be too large since the optimization is dealing with the high-dimension problem and will take a long time to run for a large h value. Here we set h as 10. $V_{i,d}$ in the expression shown in Table 1 for action *Fine Tuning* means the velocity of the d -th dimension of the personal best of i -th population. Specific expressions of $V_{i,d}$ are given as (Ji et al, 2007; Samma et al, 2016)

$$V_{i,d} = \frac{o}{it^p} u + 2B_{i,d} \quad (13)$$

where o is the acceleration factor and set as 2, p is the descent parameter that controls the decay of the velocity and is set as 20, u is a uniformly distributed random number within $[-0.5, 0.5]$, and it is the current iteration number, and

$$B_{i,d} = \begin{cases} 2V_{i,d}, & \text{if } x_{new} \prec x_{old} \\ 0.5B_{i,d}, & \text{otherwise} \end{cases} \quad (14)$$

Then the personal best can be updated in the following manner,

$$tempP_{Besti,d} = \begin{cases} P_{Besti,d} + V_{i,d}, & \text{if } x_{new} \prec x_{old} \\ P_{Besti,d}, & \text{otherwise} \end{cases} \quad (15)$$

If the temporary personal best dominates the current one, it will replace the current personal best. Otherwise, the original personal best remains unchanged. In this action process, no matter if the solution gets improved or not, there is only punishment since it takes a long time to tune each dimension. Here the punishment is set as -3 so that it cannot be called frequently for the sake of computational efficiency.

3.5 Solution procedure summary

The overall solution procedure is now summarized in Figure 3. To achieve a small yet diverse set of solutions covering the true damage scenario, we first integrate sparse population generation into MOPSO. Moreover, we develop five local search strategies based on coefficient variation and mutation, which enhance the searching capability to obtain a diverse solution set. At the same time, Q-learning is introduced

to help the particles adaptively choose the appropriate current iterative strategy for themselves. The efficacy of this new approach will be demonstrated in case studies that follow.

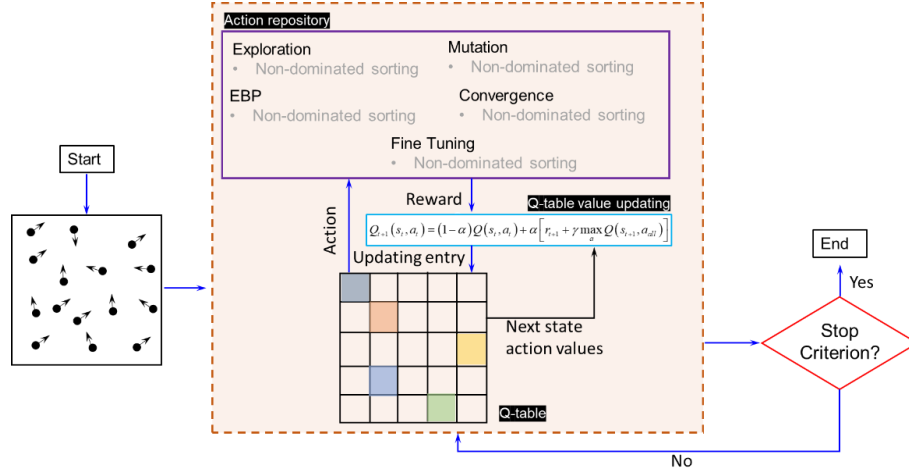


Figure 3. Flowchart of sparsity conscious reinforcement-learning guided MOPSO.

4 Damage Identification Implementation with Experimental Case Analyses

In this section, we conduct experimental case studies to verify the proposed approach. As derived in Section 2, the piezoelectric admittance change expression and sensitivity matrix are expressed in finite element forms. Here we first present the experimental setup and data acquisition, followed by finite element modeling. The numerical model and the experimental data are then employed for damage identification.

4.1 Experimental setup and data acquisition

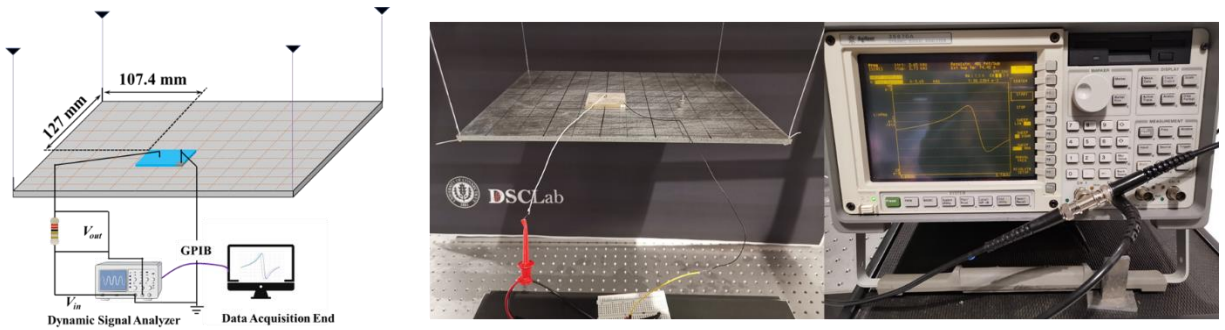


Figure 4. Schematic of experimental setup.

The experimental setup is shown in Figure 4. A piezoelectric transducer is bonded to the top surface of a plate using epoxy resin. To minimize the influence of the thickness of bonding layer, the resin layer is kept at smaller than 0.15 mm. The piezoelectric transducer is connected to an external resistor, as shown in

Figure 4. By measuring the voltage across the resistor, we can obtain the current flowing through the circuit. Thus, the admittance measurement can be obtained as $Y(\omega) = V_{\text{out}} / R / V_{\text{in}} = V_{\text{out}} / R V_{\text{in}}$ where Y is the admittance, V_{out} denotes the voltage amplitude across the resistor, and V_{in} indicates the amplitude of the input voltage. This simple yet effective measurement scheme is low-cost, and can directly facilitate the model updating as every component of the numerical model can be measured directly. A dynamic signal analyzer (Agilent 35670A) is used to acquire the frequency response. Here 5 Volts (Vpk) harmonic voltage is applied to the transducer. For each excitation frequency, 25 cycles are used to reduce the noise effect.

In damage identification using admittance measurements, the damage effect is most prominent around the resonant peaks. In this research, acquire the admittance change information around two frequencies, 3,194.95 Hz (the 28th natural frequency) and 3,705.8 Hz (the 31st natural frequency). Theoretically, one may measure admittance in even higher frequency range for higher detection sensitivity. Employing admittance measurement in higher frequencies, on the other hand, necessitates a very high dimension of finite element mesh in order for the model to converge in such frequency range, resulting in increased computational cost. Therefore, there is a trade-off. To balance the computational cost and the choice of frequency band, without loss of generality, we choose the aforementioned frequency range, which will be used for demonstration of the proposed algorithm. Further investigations on the selection of frequency range will be conducted in future. The first frequency range for admittance measurement is from 3,150 Hz to 3,230 Hz and the second is from 3,650 Hz to 3,730 Hz. Each range has 401 frequency sweeping points. The admittance curves measured under the healthy baseline state are shown in Figure 5. It is worth noting that all the data acquisitions are completed under the room temperature of 20 degrees Celsius within 0.2 degree fluctuations. The numerical simulation results are obtained after model updating, which will be explained in detail in Section 4.2 next.

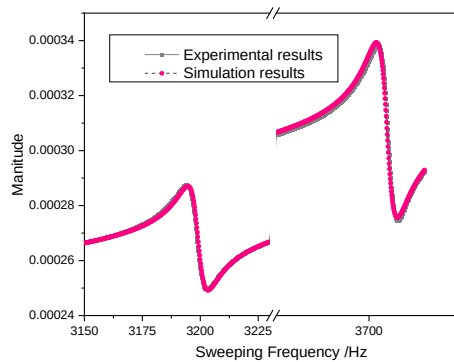


Figure 5. Admittance measurement data under the healthy state and finite element prediction after model updating.

4.2 Finite element modeling and updating

The geometry and material properties of the piezoelectric transducer and the plate are listed in Table 2. The plate is made of aluminum alloy. The piezoelectric transducer is the SM411 type from Steiner & Martins Inc. The piezoelectric transducer is placed at the following location: 107.4 mm from the left edge and 127 mm from the top edge of the plate.

Table 2. Geometry and material properties of the piezoelectric transducer and the plate.

Structure	Parameters	Values
Plate	Length (10^{-3} m)	304.8
	Width (10^{-3} m)	254
	Thickness (10^{-3} m)	3.874
	Young's modulus (10^3 MPa)	73
	Density (kg/m^3)	2769
	Poisson's ratio	0.3
Piezo Transducer (SM411)	Length (10^{-3} m)	45
	Width (10^{-3} m)	45
	Thickness (10^{-3} m)	3.5
	Young's modulus E_{11} (10^3 MPa)	74
	Young's modulus E_{33} (10^3 MPa)	54
	Density (kg/m^3)	7800
	Piezo constant g_{31} (10^{-3} m ² /C)	-10.4
	Piezo constant g_{33} (10^{-3} m ² /C)	24.2
	Permittivity constant ϵ_{33} (10^{-8} F/m)	1.348

In the numerical model, we employ the stress-voltage constitutive relation of the piezoelectric material. A finite element model is established based on an in-house MATLAB code. Both the transducer and the plate are meshed using the 20-node solid element. The total number of elements for the finite element analysis is determined as 13,766 after convergence analysis. Here the in-house MATLAB result is verified with ANSYS. In order to be consistent with the experimental setup, the admittance computed is also through the usage of a serially connected resistor with resistance 165 ohms. The stiffness and mass matrices obtained in finite element form are then substituted into the admittance expression derived in Section 2. The frequency sweeping points are the same as those used in experiment. In other words, the numerical simulation exactly matches with the experimental procedure.

It is worth noting that the material properties of the piezoelectric transducer given by the vendor are not always accurate. In addition, there are inevitable variations and uncertainties in experiment. Thus, to match the numerical results with experimental ones, model updating is conducted. For the plate structure, by measuring the total mass and volume, we can calculate its mass density. Hence there is one parameter of the plate to be updated, i.e., the Young's modulus E_{plate} . For the piezoelectric transducer, the mass density of piezoelectric transducer can similarly be obtained by measuring its mass and volume.

Furthermore, an LCR multimeter can gauge the piezoelectric capacitance (C_p) after the transducer is bonded to the plate. Subsequently, for this rectangular shaped transducer, the permittivity constant can be calculated as $\varepsilon_{33} = C_p \cdot d / A$, where d and A represent the thickness of the transducer and the area of electrode, respectively.

Before updating other parameters of the model, it is essential to understand the influence of each parameter on the admittance. We carry out sensitivity/perturbation analysis, and compare with experimental measurement to elucidate the boundary of parametric variation for the subsequent inverse analysis of model updating. Obviously, for a specific parameter, if the experimental data is properly encapsulated within the range of admittance prediction upon parametric perturbation, the corresponding variation percentage can be used as parametric boundary. Otherwise, we make adjustment to decide the parametric boundary. Moreover, the sensitivity analysis provides a quantifiable measure of the magnitude and direction of admittance changes in response to these parameter alterations. We recognize that variation of piezoelectric Young's modulus E_{11} results in lateral shift in the admittance curve and magnitude changes. Besides, piezoelectric constants g_{31} and g_{33} variations predominantly lead to distinct variations in magnitudes in both real and imaginary parts. The variation of piezoelectric Young's modulus E_{33} , on the other hand, yields much less significant effect. This insight streamlines our optimization process, enabling a focus solely on influential parameters. i.e., E_{11} , g_{31} and g_{33} . Finally, the intervals for parameter changes in model updating are decided as $\pm 20\%$ for plate Young's modulus E_{plate} and piezoelectric transducer Young's modulus E_{11} , and $\pm 35\%$ for piezoelectric constants g_{31} and g_{33} . In model updating, our goal is to minimize the difference between the admittance from the experiment ($y_h^{\text{exp}}(\omega)$) and the admittance from the numerical model ($y_h^{\text{pred}}(\omega)$). Here, the subscript 'h' refers to the healthy state. The superscripts 'exp' and 'pred' refer to the data from experiment and from model prediction, respectively. The objective function for model updating is $obj = \|y_h^{\text{exp}}(\omega) - y_h^{\text{pred}}(\omega)\|_2$. There is a total of 172 iteration steps for minimizing the objective function in model updating by adopting single-objective particle swarm optimizer. Finally, the key parameters are updated as $E_{\text{plate}} = 72.21 \text{ GPa}$, $E_{11} = 81.4 \text{ GPa}$, $g_{31} = -6.97 \times 10^{-3} \text{ m}^2 / \text{C}$, and $g_{33} = 24.2 \times 10^{-3} \text{ m}^2 / \text{C}$, as shown in Table 2.

The updated admittance curves are compared to experimental results, as shown in Figure 5. As can be observed, they match very well.

4.3 Case study setup

The plate structure to be monitored is divided into 120 segments, which means there are 120 variables to be identified in the damage index vector. It is worth emphasizing that, mathematically, the rows of the

sensitivity matrix are not necessarily linearly independent, no matter how many sweep frequency points we choose and how these sweep frequency points. Therefore, the inverse problem shown in Equation (6) is generally under-determined. This is the fundamental reason we resort to a multi-objective optimization formulation to find the solution set. Four case studies are considered to verify systematically the proposed algorithm for damage identification. Without loss of generality, to avoid altering the boundary conditions throughout the experiment that may cause unwanted variations, damage scenarios are emulated by adding one or two small mass blocks to the plate. The added mass causes the shift of admittance curves, which is equivalent to stiffness reduction. In our experiment, after we introduce the added mass, we extract admittance curve in the frequency range of interest. We then adjust, in the numerical model, the stiffness (i.e., reducing the Young's modulus of the elements) in the specific segment to which the mass is added, such that the admittance curve calculated matches the experimental one (with added mass). The percentage of stiffness reduction in the damaged segment can then be obtained numerically.

- **Case 1:** a 1.0 g mass block is attached to the 75th segment, which causes an equivalent 14.8% stiffness reduction of that segment.
- **Case 2:** a 0.6 g mass block is attached to the 75th segment, which leads to an equivalent 11.0% stiffness reduction of that segment.
- **Case 3:** a 0.6 g mass block is attached to the 116th segment, which results in an equivalent 9.4% stiffness reduction.
- **Case 4:** a 0.12g and a 0.1g mass block are attached to the 49th and 65th segments respectively, which are equivalent to 0.734% and 0.899% stiffness reductions respectively.

These four cases are designed with damage locations chosen randomly. The first two cases have the same location but with different damage severities. Case 3 differs from Cases 1 and 2 in location and severity. Case 4 has multiple damage locations with smaller severities to emulate early damage occurrence. Such a case study setup can demonstrate the capability of the proposed algorithm to identify damage, regardless of the damage sizes and locations. To evaluate the algorithm performance, several well-established optimization algorithms, including the MOEA/D (Zhang and Li, 2007), the original MOPSO (Coello et al, 2004), and the NSGAI (Deb et al, 2000), are also employed and compared. Besides, the classical optimization solvers are selected here since they are still popular in the community of structural health monitoring (Lin et al., 2022; Xu et al., 2018; Lin et al., 2020; Gomes et al., 2018; Cha and Buyukozturk 2015; Tiachacht et al., 2018).

4.4 Effect of sparsity generation

As mentioned, damage usually occurs in a small number of segments especially at early stage, thus resulting in a sparse damage index vector. In this sub-section, the aforementioned four cases are analyzed

to examine firstly the role that the sparsity generation algorithm plays. For this purpose, we implement the sparse population generation algorithm into MOPSO, and the resulted optimization algorithm is referred to as Sparse-MOPSO. It is worth noting that this Sparse-MOPSO is run without the reinforcement learning components, as we want to first focus on the benefit of sparsity generation. Three other optimization algorithms are employed for comparison. For a fair comparison, we set the maximum iterations as 20,000, 40,000, 60,000, 80,000, and 100,000, respectively, and investigate the results. In addition, for each algorithm, the population size is set to be 30. Each algorithm is run 10 times independently.

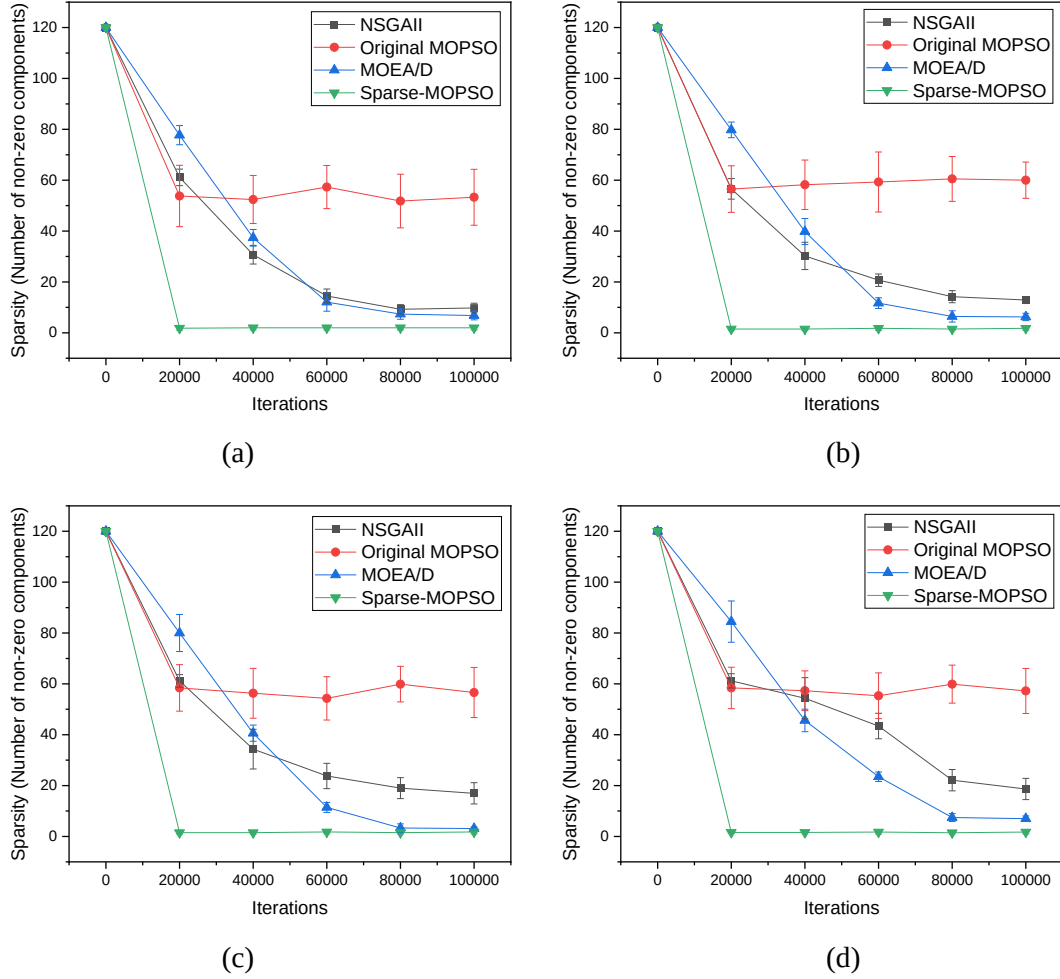


Figure 6. Comparison of number of non-zero components in damage index vector vs iteration by different optimization algorithms (a) Case 1, (b) Case 2, (c) Case 3, and (d) Case 4.

The comparison of sparsity (number of non-zero damage locations identified) versus iterations is provided in Figure 6 for the four cases mentioned. For each maximum iteration case, there are 10 independent runs. Therefore, the sparsity obtained is shown as average with error bar. All the algorithms

reach convergence after 80,000 iterations, as the number of non-zero damage locations no longer change. Comparing the Sparse-MOPSO with the original MOPSO, we can observe that the sparsity generation algorithm employed in Sparse-MOPSO can achieve sparsity by controlling the number of non-zero damage index vector components to around 3 after convergence. The number of non-zero damage locations from the original MOPSO, on the other hand, is around 60 even after it reaches convergence. This indicates that the sparsity generation algorithm introduced indeed works well in dealing with the sparse characteristics in damage identification. By comparing the Sparse-MOPSO with the NSGAI and the MOEA/D, we can observe that the Sparse-MOPSO outperforms the other two by achieving sparsity more efficiently. For example, with the maximum iterations of 20,000, the Sparse-MOPSO achieves sparsity with about 3 non-zero damage locations while that number is greater than 60 for other algorithms.

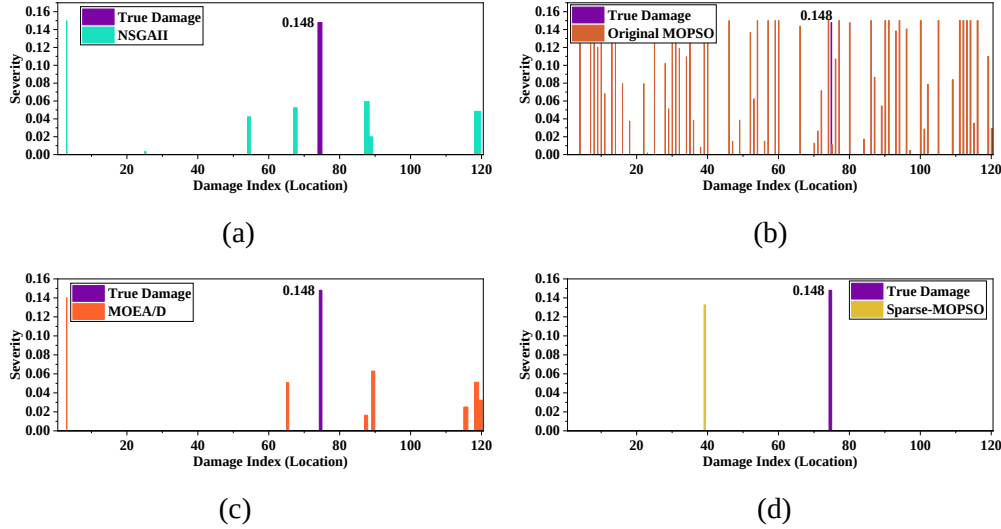


Figure 7. A representative solution from different algorithms after 100,000 iterations for Case 1 (a) NSGAI, (b) original MOPSO, (c) MOEA/D, and (d) Sparse-MOPSO.

In Figure 7, we compare one representative solution for Case 1 obtained by different algorithms after 100,000 iterations. The number of columns is the number of damage locations identified, and the height of a column represents the percentage of stiffness reduction at the specific damage location (i.e., the segment number). In this case, since the true damage scenario is located at only one segment, the expectation for all algorithms is that they can achieve the identification of one damage location after convergence. However, from the results, only the Sparse-MOPSO algorithm can find one damage location, although it does not identify the true location correctly. It is worth noting that at this point, we are more concerned about the performance of sparsity because only the sparsity generation is integrated into the Sparse-MOPSO. The solution correctness will be evaluated after we activate the reinforcement learning part of the MOPSO. For

the NSGAI and the MOEA/D, the number of damage locations identified is around 7 after the respective convergence. For the original MOPSO, the identified damage locations are spread over almost the entire structure, thus it does not provide meaningful information for damage identification. It is thus evident that the sparsity generation algorithm proposed in this research works well in dealing with the sparse feature in damage identification.

4.5 Examination of reinforcement learning guided local search

In the preceding sub-section, the results indicate the necessity of the sparsity generation algorithm in handling sparse characteristics in damage identification. However, from the representative solution obtained, one can see that the solution from the Sparse-MOPSO, which yields the sparse solution set, does not capture the true damage scenario. The untrue solution represents a local minimum where the algorithm is trapped. To tackle this challenge, the local searching strategies combined with Q-learning developed in Section 3 are now activated. Hereafter, the proposed sparsity-conscious MOPSO guided by reinforcement-learning is referred to as SRL-MOPSO.

The four cases mentioned above are now thoroughly examined. For a truly fair comparison, we introduce the sparsity generation algorithm into all algorithms to be compared as well, including the NSGAI, the MOPSO, and the MOEA/D. The settings for all algorithms are consistent: maximum iterations 100,000 and population 30. Other related algorithmic parameters are provided in Section 3. The solutions obtained for Cases 1 to 4 are plotted in Figures 8 to 11. Within the multi-objective optimization formulation for inverse analysis, multiple solutions are obtained for the under-determined damage identification problem. In these figures, each sub-plot represents one solution obtained from an individual algorithm. As explained, the number of columns in each subplot represents the damage locations identified, and the height of a column is the corresponding severity value. All the solutions are arranged, from left to right, with the increase of the number of identified damage locations.

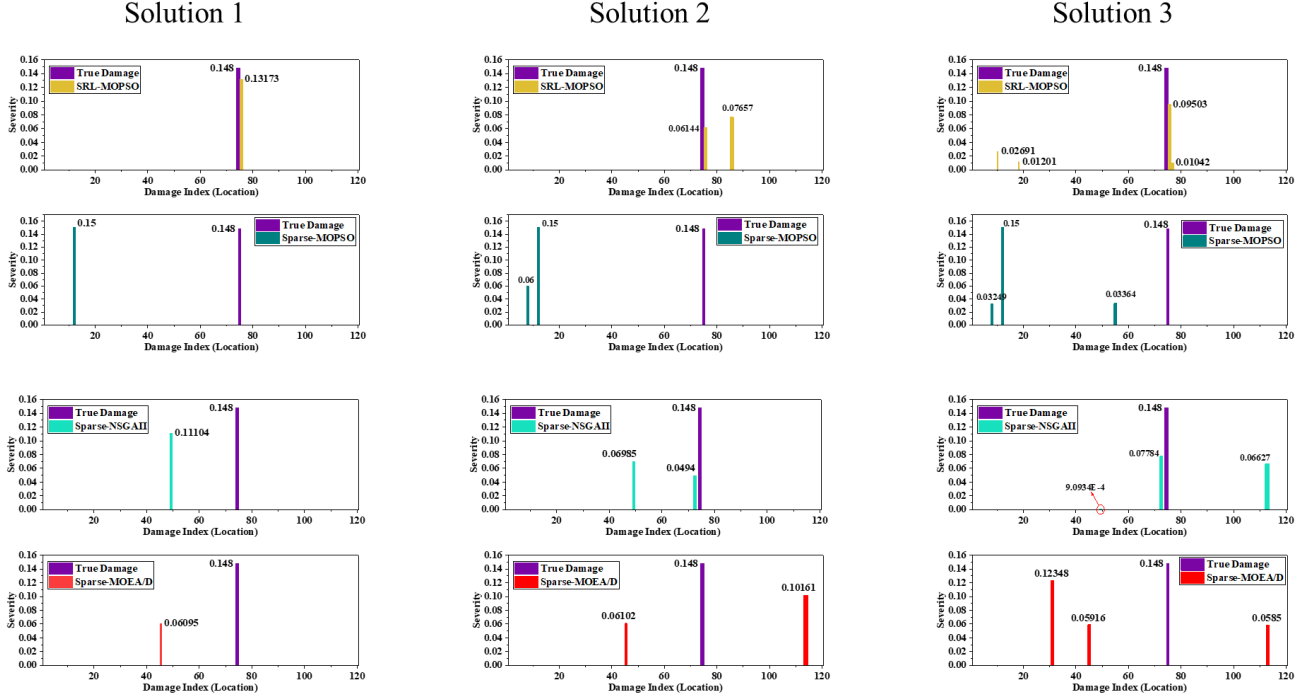


Figure 8. Case 1 identification results from various algorithms.

For Case 1, the solutions obtained from various algorithms are shown in Figure 8. Since the sparse generation is now introduced to all algorithms, we can observe that all solutions achieve a good sparsity. The maximum number of identified damage locations in Case 1 is 4, which is indeed small as compared to the total number of possible damage locations, i.e., 120. This indicates that the sparse population generation algorithm is a generic strategy that can enable the general optimization algorithms to improve their capabilities for damage identification. The difference between the SRL-MOPSO and the Sparse-MOPSO is that the former is integrated with local searches combined with Q-learning algorithm which aims at enhancing the global search capability toward finding the true damage scenario. By comparing the solutions obtained from the SRL-MOPSO and the Sparse-MOPSO (i.e., the first two rows in Figure 8), we can find that the 3 solutions generated by the SRL-MOPSO include 1, 2 and 4 damage locations, respectively, and all these solutions point to the same location, the 75th segment, that aligns with the true damage location. Indeed, the first solution from the SRL-MOPSO indicates one damage location at the 75th segment with stiffness reduction 13.173%, which matches well with the true damage scenario, i.e., the 75th segment with 14.8% stiffness reduction. Although the second and the third solutions from the SRL-MOPSO point to additional damage locations besides the true damage location, all the additional damage locations identified have quite small severity values. Collectively, these 3 solutions identified provide very consistent identification results that can be utilized in decision making in engineering practice. In contrast, the Sparse-MOPSO generally give solutions that are very different from the true damage scenario. Indeed, none of the

3 solutions from the Sparse-MOPSO points to the true damage location. Similarly, the solutions obtained from Sparse-NSGAI and the Sparse-MOEA/D all exhibit significant discrepancy with respect to the true damage scenario. This means that these algorithms may get trapped in local minima that are different from the true damage scenario.

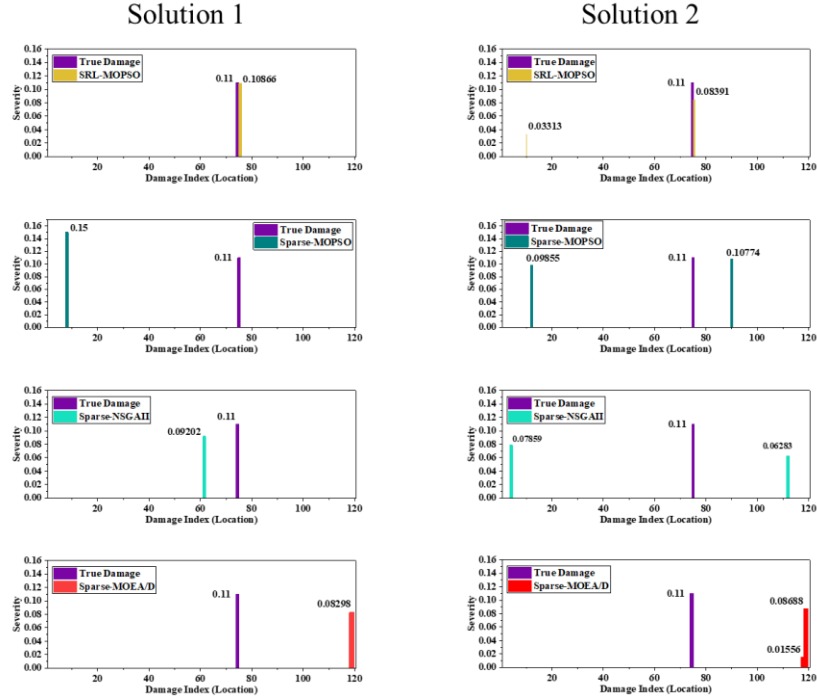


Figure 9. Case 2 identification results from various algorithms.

In Case 2, the true damage scenario has the same location at the 75th segment but different severity with respect to Case 1. The solutions obtained from various algorithms are plotted in Figure 9. In this case, the maximum number of non-zero damage locations identified by all algorithms is 2, after we integrate the sparse population generation. However, the Sparse-MOPSO, the Sparse-NSGAI and the Sparse-MOEA/D produce solutions that all deviate from the true damage scenarios. In comparison, the SRL-MOPSO is enhanced by the local search strategies which can help the particles to escape from the local optima and thus achieve a better global search. This leads to solutions that are much better matching with the true damage scenario. For the two solutions obtained from the SRL-MOPSO, we can see that the number of identified damage locations are 1 and 2, respectively, and both solutions point to the true damage location, i.e., the 75th segment. For solution 1 obtained from the SRL-MOPSO, the relative error between the true damage severity 11% stiffness reduction and the identified result 10.866% is only 1.22%, indicating excellent accuracy. For solution 2, the additional damage location identified has quite small severity level. Therefore this second solution is only a minor deviation from the true damage scenario.

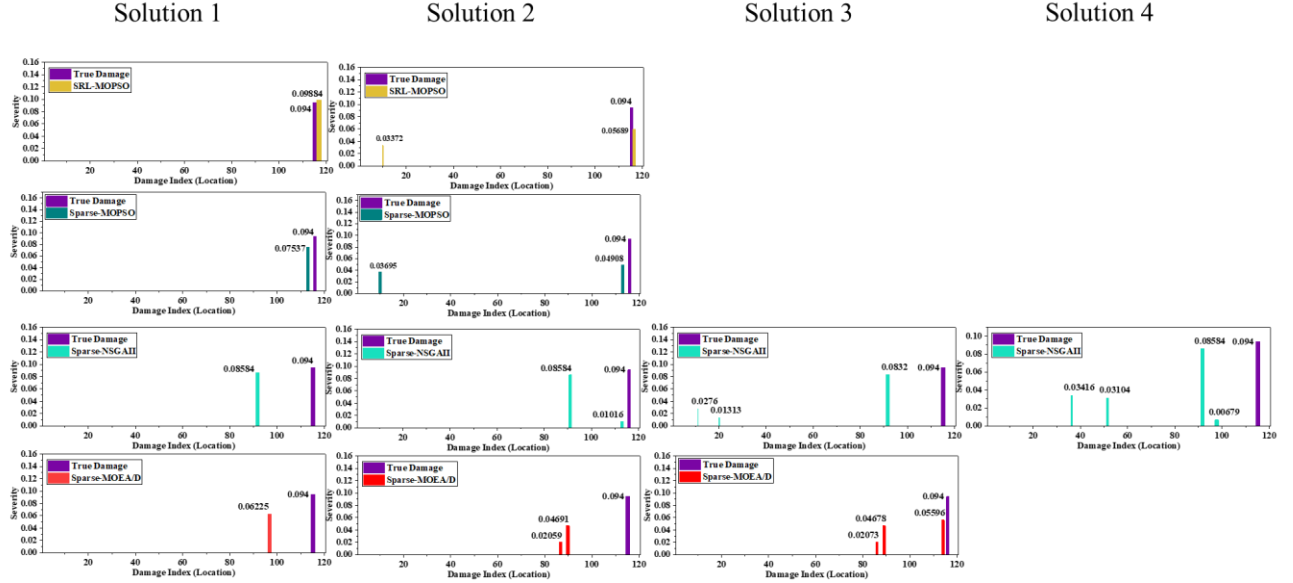


Figure 10. Case 3 identification results from various algorithms.

In Case 3, the damage occurs at the 116th segment, i.e., a different location than in Cases 1 and 2. The solutions obtained from various algorithms are shown in Figure 10. Owing to the integration of sparse population generation, the maximum number of non-zero damage locations identified is 4 by the Sparse-NSGAI. Both the SRL-MOPSO and the Sparse-MOPSO produce 2 solutions. Both solutions from the Sparse-MOPSO, however, point to incorrect damage locations. The other two algorithms, the Sparse-NSGAI and the Sparse-MOEAD, identify 4 solutions and 3 solutions, respectively. Interestingly, none of these solutions points to the true damage location. The damage scenarios identified by all the other algorithms do not match with the true damage scenario. Once again, the SRL-MOPSO can identify the true damage scenario, shown as its solution 1. In its solution 2, although two damage locations are identified, the solution points to the 116th segment as a main location, which fits the true damage scenario. It is worth noting that modeling uncertainties and measurement noise inevitably exist. As such, it is expected that some solutions identified through the multi-objective formulation are different from the true damage scenarios. The SRL-MOPSO can identify a small solution set that includes the true damage scenario.

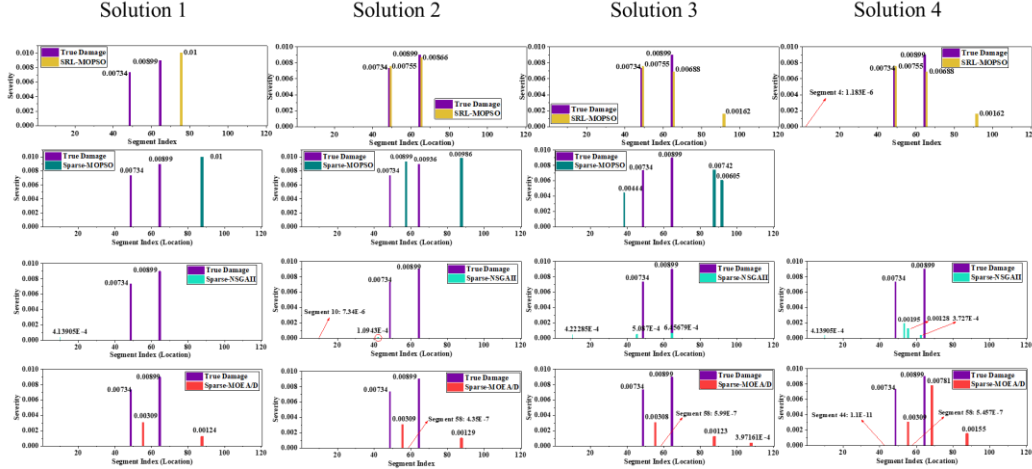


Figure 11. Case 4 identification results from various algorithms.

In Case 4, the damage is located at the 49th and the 65th segments with 0.734% and 0.899% stiffness reductions, respectively. This case setup aims at examining the capability of the algorithms when damage occurs at multiple locations. In this particular case, to challenge the algorithms, the damage severities are designed to be quite small. The solutions obtained from different algorithms are plotted in Figure 11. The number of solutions identified by the SRL-MOPSO, the Sparse-MOPSO, the Sparse-NSGAI and the Sparse-MOEAD are, respectively, 4, 3, 4, and 4. Observing the first solution identified by the SRL-MOPSO, one can observe that it indicates a single damage, which does not match the true damage scenario with two damage locations. This is not a surprise, because one of the objective functions in the multi-objective optimization is the sparsity of the damage index vector, and the algorithm may indeed find a solution with single damage. Similarly, the first solutions identified by the other three algorithms do not match with the true damage scenario either. Meanwhile, comparing the first two rows of Figure 11, one can see that the SRL-MOPSO clearly outperforms the Sparse-MOPSO, as the former yields its solutions 2 to 4 that all are very close to the true damage scenarios. Amongst them, solution 2 with two damage locations basically matches with the true damage scenario in both location and severity. Although solutions 3 and 4 point to other damage locations in addition to the locations of the true damage scenario, those additional locations all have very small severities. Clearly, solutions 2 to 4 are quite consistent, and again can provide import information for decision making in engineering practice. In comparison, the Sparse NSGAI and the Spare-MOEAD are unable to produce solutions that are close to the true damage scenario. This further highlights the importance of incorporating the strategies outlined in Section 3 for solution diversity and accuracy.

In summary, the case studies demonstrate the performance of the proposed approach. A small solution set that always includes the true damage scenario is obtained for each case. In actual practice, once multiple

solutions are identified, there will subsequently be a decision-making process. If all these solutions point to insignificant damage, continuing operation of the structure can be recommended. If one or several solutions point to potentially severe consequence, additional measurements using different sensing techniques including human inspection can be dispatched to further investigate this small set of possible damage scenarios. On the other hand, if the solution set contained a very large number of solutions, then it would lose the essence of structural health monitoring because an operator would then need to examine too many possible damage scenarios afterwards. Obviously, the proposed SML-MOPSO can indeed fulfill the task of identifying a small solution set that always includes the true damage scenario.

4.6 Effect of Q-table guided action selection and computational cost analysis

In this sub-section, we analyze the effect of Q-learning by examining the interaction between the Q-table and action selection, and also discuss the computational cost and overhead.

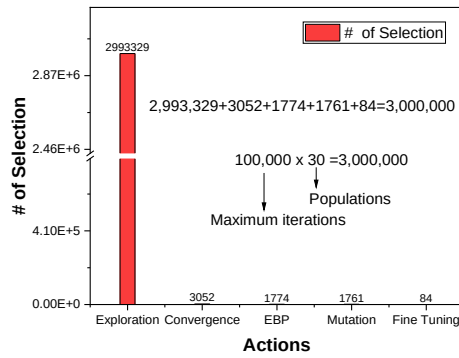


Figure 12 Number of selections of actions guided by Q-table (Case 1).

As stated in Section 3.3, an action is selected based on the maximum Q value in Q-table. Therefore, attention is paid to all the actions that are selected and executed after convergence. As shown in Figure 12, the histogram counts the number of times each local search strategy is executed after the algorithm converges. For those algorithms without local search, they tend to fall into local optima and thus reach premature convergence. This results in quite significant unexplored space in search in which the global optimum may reside. On the other hand, the proposed SRL-MOPSO takes most of the computational effort on the *Exploration* strategy to explore the entire parametric space. Of the four remaining local search strategies, the *Convergence* is executed nearly 3,000 times. This indicates that the particles are flying to the global optimum, i.e., reaching convergence, while exploring the unknown space. In comparison, both *EBP* and *Mutation* are executed nearly 1,700 times, indicating that they play an almost equal role. It is worth noting that the last strategy *Fine Tuning* is executed only 84 times. There are two possible reasons. One is

that we only give a penalty to this strategy regardless of good or bad solution obtained after this strategy is executed since it takes up a particularly large amount of time when dealing with a high-dimensional damage index vector. Thus, it has a lower probability of being selected. The other reason is that when initializing the Q-table, we set Q-table value of -5 for the *Fine Tuning* and give priority for other local searches to be selected. The fact that the *Fine Tuning* strategy is executed such infrequently does not mean it is not needed. In fact, it is because of this strategy that other strategies have led to better performance. Without loss of generality, here we report the results based on Case 1 computation. Similar observations are obtained for other cases. The interaction between Q-table and action selection is animated as a gif file (Appendix).

Table 3 Computational cost/performance comparison between MOPSO and SRL-MOPSO

Metaheuristics	20,000 iterations			100,000 iterations		
	Time (s)	sparsity	convergence	Time (s)	Sparsity	convergence
MOPSO	179	56.7	No	893	53.3	yes
SRL-MOPSO	258	1.8	Yes	1091	1.8	yes

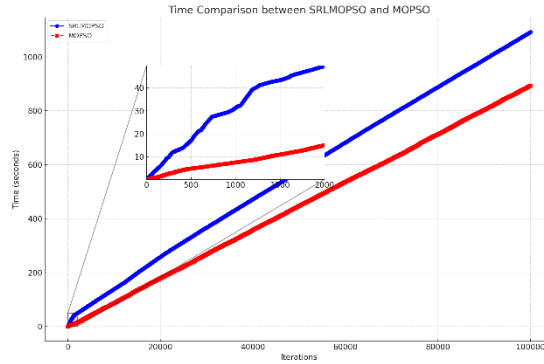


Figure 13. Elapsed time comparison with and without integration of Q-learning.

Finally, we analyze the computational cost with and without Q-learning. In what follows we present a comparison study. Throughout this research, damage identification inverse analysis is carried out on a regular Dell computer with Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz (2 processors) a 128 GB RAM. Table 3 shows a comparative analysis between the MOPSO and the SRL-MOPSO algorithms, detailing their respective computational time, sparsity levels, and convergence statuses over two specific iteration counts: 20,000 and 100,000. Figure 13 illustrates the elapsed computational time for both algorithms across a broader range of iterations, highlighting the differences in time efficiency as the number of iterations increases. The proposed SRL-MOPSO, which incorporates Q-learning, understandably introduces a computational overhead. Specifically, for 20,000 iterations, the SRL-MOPSO takes 79 seconds longer than

MOPSO, and for 100,000 iterations, it takes an additional 198 seconds. However, the benefits brought about by this integration far outweigh the time increase. Most notably, the sparsity improvement is drastic, dropping from 56.7 to 1.8 at 20,000 iterations and maintaining this enhanced sparsity at 100,000 iterations, as shown in Figure 6. Moreover, while the MOPSO fails to converge at 20,000 iterations, the SRL-MOPSO achieves this significant milestone, demonstrating its enhanced capability. In essence, the SRL-MOPSO's added computation time is only marginal, while leading to a reasonable trade-off for the pronounced advancements in performance.

5 Conclusion

In this research, a new sparsity conscious multi-objective optimization framework is developed to fully unleash the capability of piezoelectric admittance measurement measured in high-frequency range for structural damage identification. While promising, the measurement information of piezoelectric admittance change information is generally limited, oftentimes leading to an under-determined problem for damage identification. To tackle this challenge, multi-objective optimization inverse analysis is formulated. The multi-solution nature of such a formulation fits naturally the structural health monitoring practice. Specifically, we focus on the development of effective algorithm that can lead to a small solution set that includes the true damage scenario. We choose the multi-objective particle swarm optimization (MOPSO) as the basic platform and incorporate metaheuristics to enhance its capability toward damage identification. To obtain a small yet diverse solution set, a sparsity generation algorithm is integrated into MOPSO. Furthermore, a series of local search strategies through coefficient variation-based and mutation-based techniques are synthesized that form an action repository to enhance the global searching capability of MOPSO. To guide the particle to adaptively select the local search, Q-learning is integrated. Case studies utilizing physical experiment of piezoelectric active interrogation through piezoelectric admittance measurement and the associated finite element modeling are conducted. The importance of sparse population generation is first examined, followed by damage identification performance investigation. The proposed sparsity-conscious reinforcement learning guided MOPSO (SRL-MOPSO) is compared with benchmark algorithms including the NSGAI, the MOEA/D, and the original MOPSO. In all the cases examined utilizing experimental data, the SRL-MOPSO outperforms other algorithms, and shows excellent results in identifying a small solution set that includes the true damage scenario. This framework can potentially be extended to other model-based structural health monitoring approaches.

Appendix

Visualization of Q-learning process is provided as an animation in gif format, where the heatmap shows how the action and state are selected and updated. <https://github.com/yantzan/Animation-of-Q-table>

Data availability

All the algorithms and analyses are implemented through MATLAB. All related data including simulation data and experimental data are available from the corresponding author upon request.

Acknowledgment

This research is supported in part by a Space Technology Research Institutes grant (number 80NSSC19K1076) from NASA's Space Technology Research Grants Program and in part by the National Science Foundation under grant CMMI-1825324.

References

- Alkayem, N.F., Cao, M., Zhang, Y., Bayat, M. and Su, Z., Structural damage detection using finite element model updating with evolutionary algorithms: a survey. *Neural Computing and Applications*, 30 (2018), 389-411.
- Cao, P., Shuai Q., and Tang J., Structural damage identification using piezoelectric impedance measurement with sparse inverse analysis. *Smart Materials and Structures* 27(3) (2018) 035020.
- Chen, Z. and Sun, H., 2021. Sparse representation for damage identification of structural systems. *Structural Health Monitoring*, 20(4) (2021) 1644-1656.
- Cha, Y.J. and Buyukozturk, O., Structural damage detection using modal strain energy and hybrid multiobjective optimization. *Computer-Aided Civil and Infrastructure Engineering*, 30(5) (2015) 347-358.
- Coello, C.A.C., Pulido, G.T. and Lechuga, M.S., Handling multiple objectives with particle swarm optimization. *IEEE Transactions on evolutionary computation*, 8(3) (2004) 256-279.
- Ding, Z., Li, J. and Hao, H., Non-probabilistic method to consider uncertainties in structural damage identification based on Hybrid Jaya and Tree Seeds Algorithm. *Engineering Structures*, 220 (2020) p.110925.
- Ding, Z., Hou, R. and Xia, Y., Structural damage identification considering uncertainties based on a Jaya algorithm with a local pattern search strategy and L0.5 sparse regularization. *Engineering Structures*, 261 (2022), 114312.
- Ding, Z., Li, J. and Hao, H., Structural damage identification using improved Jaya algorithm based on sparse regularization and Bayesian inference. *Mechanical Systems and Signal Processing*, 132 (2019) 211-231.
- Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T., A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Parallel Problem Solving from Nature PPSN VI: 6th*

International Conference Paris, France, September 18–20, 2000 Proceedings 6 (pp. 849-858). Springer Berlin Heidelberg.

Dash, M. and Liu, H., Feature selection for classification. *Intelligent data analysis*, 1(1-4) (1997) 131-156.

Davenport, M.A., Duarte, M.F., Eldar, Y.C. and Kutyniok, G., *Introduction to compressed sensing*. (2014) 1-64

Fan, X. and Li, J., Damage identification in plate structures using sparse regularization based electromechanical impedance technique. *Sensors*, 20(24) (2020), 7069.

Fan, X., Li, J. and Hao, H., Impedance resonant frequency sensitivity based structural damage identification with sparse regularization: experimental studies. *Smart Materials and Structures*, 28(1) (2018), 015003.

Gad, A.G., Particle swarm optimization algorithm and its applications: a systematic review. *Archives of computational methods in engineering*, 29(5) (2022) 2531-2561.

Gomes, G.F., Mendéz, Y.A.D., da Cunha, S.S. and Ancelotti, A.C., A numerical–experimental study for structural damage detection in CFRP plates using remote vibration measurements. *Journal of Civil Structural Health Monitoring*, (2018) 8, 33-47.

Ji, Z., Liao, H., Wang, Y. and Wu, Q.H., A novel intelligent particle optimizer for global optimization of multimodal functions. In *2007 IEEE Congress on Evolutionary Computation* (pp. 3272-3275). IEEE.

Jin, S.S., Cho, S., Jung, H.J., Lee, J.J. and Yun, C.B., A new multi-objective approach to finite element model updating. *Journal of Sound and Vibration*, 333(11) (2014) 2323-2338.

Kennedy, J. and Eberhart, R., Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948) (1995, November). IEEE.

Kim, Jinki, and Wang, K.W., An enhanced impedance-based damage identification method using adaptive piezoelectric circuitry, 23(9) (2014) 095041.

Lim, W.H. and Isa, N.A.M., Two-layer particle swarm optimization with intelligent division of labor. *Engineering Applications of Artificial Intelligence*, 26(10) (2013) 2327-2348.

Li, H., Ai, D., Zhu, H. and Luo, H., An Orthogonal Matching Pursuit based signal compression and reconstruction approach for electromechanical admittance based structural health monitoring. *Mechanical Systems and Signal Processing*, 133 (2019), 106276.

Lin, T.Y., Tao, J. and Huang, H.H., A multiobjective perspective to optimal sensor placement by using a decomposition-based evolutionary algorithm in structural health monitoring. *Applied Sciences*, (2020) 10(21), 7710.

Li, X., Niu, S., Bao, H. and Hu, N., Improved Adaptive Multi-Objective Particle Swarm Optimization of Sensor Layout for Shape Sensing with Inverse Finite Element Method. *Sensors*, (2022) 22(14), 5203.

Meerza, S.I.A., Islam, M. and Uzzal, M.M., Q-Learning Based Particle Swarm Optimization Algorithm for Optimal Path Planning of Swarm of Mobile Robots. In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) (pp. 1-5). IEEE.

Minh, H.L., Khatir, S., Wahab, M.A. and Cuong-Le, T., An Enhancing Particle Swarm Optimization Algorithm (EHVPSO) for damage identification in 3D transmission tower. *Engineering Structures*, 242 (2021), 112412.

Pereira, J.L.J., Francisco, M.B., de Oliveira, L.A., Chaves, J.A.S., Cunha Jr, S.S. and Gomes, G.F., Multi-objective sensor placement optimization of helicopter rotor blade based on Feature Selection. *Mechanical Systems and Signal Processing*, 180 (2022), 109466.

Park, G., Sohn, H., Farrar, C.R. and Inman, D.J., Overview of piezoelectric impedance-based health monitoring and path forward. *Shock and vibration digest*, 35(6) (2003) 451-464.

Qian, C., Yu, Y. and Zhou, Z.H., Subset selection by Pareto optimization. *Advances in neural information processing systems*, 28 (2015).

Shuai, Q., Zhou, K., Zhou, S. and Tang, J., Fault identification using piezoelectric impedance measurement and model-based intelligent inference with pre-screening. *Smart Materials and Structures*, 26(4) (2017), 045007.

Shi, Binkai, Maosen Cao, Zijian Wang, and Wieslaw Ostachowicz. A directional continuous wavelet transform of mode shape for line-type damage detection in plate-type structures. *Mechanical Systems and Signal Processing* 167 (2022), 108510.

Samma, H., Lim, C.P. and Saleh, J.M., A new reinforcement learning-based memetic particle swarm optimizer. *Applied Soft Computing*, 43 (2016) 276-297.

Tiachacht, S., Bouazzouni, A., Khatir, S., Wahab, M.A., Behtani, A. and Capozucca, R., Damage assessment in structures using combination of a modified Cornwell indicator and genetic algorithm. *Engineering Structures*, 177 (2018) 421-430.

Tian, Y., Zhang, X., Wang, C. and Jin, Y., An evolutionary algorithm for large-scale sparse multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 24(2) (2019) 380-393.

Ventresca, M., Harrison, K.R. and Ombuki-Berman, B.M., The bi-objective critical node detection problem. *European Journal of Operational Research*, 265(3) (2018) 895-908.

Wang, X. and Tang, J., Damage identification using piezoelectric impedance approach and spectral element method. *Journal of Intelligent Material Systems and Structures*, 20(8) (2009) 907-921.

Watkins, C.J. and Dayan, P., Q-learning. *Machine learning*, 8 (1992) 279-292.

Xu, C. and Deng, M., Lamb wave imaging based on multi-frequency sparse decomposition. *Mechanical Systems and Signal Processing*, 174 (2022), 109076.

Xue, B., Zhang, M. and Browne, W.N., Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE transactions on cybernetics*, 43(6) (2012) 1656-1671.

Xu, Y., Ding, O., Qu, R. and Li, K., Hybrid multi-objective evolutionary algorithms based on decomposition for wireless sensor network coverage optimization. *Applied Soft Computing*, (2018) 68, 268-282.

Yao, R., Pakzad, S.N. and Venkitasubramaniam, P., Compressive sensing based structural damage detection and localization using theoretical and metaheuristic statistics. *Structural Control and Health Monitoring*, 24(4) (2017), p.e1881.

Zhu, J., Qing, X., Liu, X. and Wang, Y., Electromechanical impedance-based damage localization with novel signatures extraction methodology and modified probability-weighted algorithm. *Mechanical Systems and Signal Processing*, 146 (2021), 107001.

Zheng, T., Luo, W., Liang, X. and Tong, H., Damage identification for functionally graded materials using improved experience-based learning algorithm based on sparse regularization. *Engineering Optimization*, (2022) 1-20.

Zhang, Q. and Li, H., MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6) (2007) 712-731.