Harnessing Collaborative Learning Automata to Guide Multi-objective Optimization based Inverse Analysis for Structural Damage Identification

Yang Zhang, Ph.D. Student
School of Mechanical, Aerospace and Manufacturing 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
School of Mechanical, Aerospace and Manufacturing Engineering
University of Connecticut
191 Auditorium Road, Unit 3139, Storrs, CT 06269, USA
Phone: +1 (860) 486-5911; Email: jiong.tang@uconn.edu

Applied Soft Computing
(Accepted)

† Corresponding author

Harnessing Collaborative Learning Automata to Guide Multi-objective Optimization based Inverse Analysis for Structural Damage Identification

Yang Zhang[1], Kai Zhou[2], Jiong Tang[†1]

1 School of Mechanical, Aerospace and Manufacturing Engineering, University of Connecticut

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

Abstract

Structural damage identification based on physical models is often transformed into an optimization problem that minimizes the difference between measurement information of structure being monitored and the model prediction in the parametric space. However, the objective function in this context often exhibits multimodality, involving high-dimensional variables due to the reliance on finite element models for damage identification. These features pose challenges to optimization algorithms, where entrapment in local solutions can lead to false positives and false negatives in damage identification. In this research, we propose a reinforcement learning based multi-swarm optimizer to tackle such challenges in pursuit of a small yet diverse solution set that can capture the true damage scenario as one of the solutions. The proposed method leverages the flexibility of the particle swarm optimizer and incorporates novel strategies of metaheuristics to realize targeted improvement. To enable the particle swarm to adaptively select the appropriate search strategy based on the current environment, we adopt the learning automata technique, which sidesteps the need for reward strategy selection that is usually *ad hoc* at each step of the search. The integration harnesses the automatic learning and self-adaptation capabilities of learning automata, enabling the particles to navigate based on environmental signals. This leads to accumulated probabilities tied to advantageous movements, fostering an adaptive exploration of particles in the search space. The proposed approach is first validated through implementing into benchmark test cases with comparisons. It is then applied to structural damage identification with piezoelectric admittance experimental signals. `The results highlight the capability of the algorithm to identify a small solution set with high accuracy to match the actual damage scenario.

Keywords: multi-objective particle swarm optimization, reinforcement learning, learning automata, multimodality, damage identification.

[†] Corresponding author

1 Introduction

Structural Health Monitoring (SHM) encompasses techniques and processes aimed at assessing the integrity and performance of various structures in aerospace, marine, and manufacturing and transportation systems, highlighting its critical role in ensuring safety and longevity. The necessity of SHM stems from the ever-present risk of damage which can compromise structural functionality and safety. Damage identification within SHM is crucial, as it enables early detection and remediation, thereby mitigating risks and reducing maintenance costs. Typically, damage identification is reframed as an inverse optimization problem, where the goal lies in reconciling the real dynamic responses of the current structure, such as frequencies and mode shapes (Zhang and Sun 2021; Avci et al, 2021; Zhou and Tang 2021; Ding et al, 2022), or piezoelectric impedance (Fan et al, 2019; Cao et al, 2023a; Zhang et al, 2024a), with predictions derived from finite element models. These responses are intrinsically linked to structural properties like mass, stiffness, and damping. This inverse problem can be addressed through meta-heuristic optimization algorithms. However, compared with the number of unknowns, the measurement information is quite limited. As such, structural damage identification is oftentimes an under-determined problem. Fortunately, expert knowledge can be used to tackle the underdetermination, i.e., damage usually affects a limited number of areas within a structure, or expressed in mathematical terms, structural damage manifests sparsely. Thus, the regularization is typically utilized by adding a l_0 or l_1 norm to the above residual formulation with a regularization weight. Although advancements have been made, single-objective optimization yields a single solution which may be contaminated by model mismatch and measurement noise, and the inclusion of regularization in these approaches requires the selection of weighting constants that are challenging to determine *a priori*. Alternatively, structural damage identification can be formulated as a multi-objective optimization problem that minimizes concurrently a) the difference between measurements and model prediction in damage parametric space; and b) the l_0 norm of the damage index vector. This has multiple advantages intuitively. It avoids the *ad hoc* selection of weights between individual objectives employed in single-objective optimization. Moreover, this formulation yields multiple solutions, which fits the nature of structural damage identification. Yet, the objective function in this context often exhibits multimodality, involving high-dimensional variables due to the reliance on finite element models for damage identification. This complexity can lead to local optima, which, if the optimization process gets trapped, may result in false identifications, or missed damages, posing significant challenges for the optimization algorithms used in structural damage identification.

Given the intricate nature of the optimization challenges outlined, it becomes imperative to explore tailored algorithmic solutions. In light of the "No Free Lunch" principle, which posits that no single algorithm can optimally solve all optimization problems, the customization of algorithms based on the specific settings and requirements of the optimization model becomes imperative. This necessitates tailoring algorithms to enhance their performance, making the choice and adaptation of algorithms a pivotal aspect of addressing the challenges inherent in structural damage identification. Typically, the improvement of algorithms is grounded on foundational evolutionary optimization techniques, such as genetic algorithms (Li et al, 2023), Jaya algorithm (Ding et al, 2022), and particle swarm optimization (Zhang et al, 2024a), among others. Guided by a problem-oriented approach, the design of specific local search strategies or the integration of other local search algorithms is often employed to augment the performance of these foundational algorithms. For example, Livani et al (2018) improved flaw detection in two-dimensional structures by integrating the extended spectral finite element method (XSFEM) with a particle swarm optimization (PSO) algorithm, enhanced by a novel active/inactive flaw (AIF) strategy inspired by earthquake engineering, to efficiently locate cracks and holes while reducing computational costs. Ding et al (2022) enhanced Jaya algorithm for structural damage identification, integrating one-step K-means clustering, Hooke–Jeeves pattern search, and linear population reduction to tackle optimization challenges and uncertainties. Thanh et al (2022) improved the grey wolf optimizer by incorporating Lévy flight and new leadership dynamics, improving convergence and effectiveness in structural damage identification. Li et al (2023) proposed a novel method through the integration of an advanced surrogate modeling technique, specifically the sparse polynomial chaos expansion model, with a hybrid optimization strategy combining K-means clustering and genetic algorithm.

The improvements so far have predominantly focused on the application of a singular strategy to enhance algorithmic performance. However, the optimization problems in structural damage identification usually invite the development and incorporation of multiple local search strategies to balance the exploration and exploitation. This raises an intriguing question: how does one select the most appropriate strategy at each iteration to effectively navigate the search space? The selection process itself becomes a critical factor in the optimization journey, as the adaptability of algorithms to the shifting landscape of the problem at hand is key to their success. Recent explorations have seen the fusion of multiple local search strategies with machine learning techniques, like reinforcement learning, which can intelligently adjust their search adaptively based on the reinforcement signals. Among the efforts, Samma et al (2016) introduced multiple local strategies, namely, exploration, convergence, and various jump and tuning operations, within a particle swarm optimization framework, employing Q-learning for strategy selection, thereby enhancing algorithm performance on benchmark optimization problems. Ding et al (2023) developed a global vibration-based method utilizing an innovative Q-learning evolutionary algorithm, which combines K-means clustering, Jaya, and tree seeds algorithms, for the first time to accurately detect and quantify debonding in FRP strengthened structures through modal testing and model updating techniques, enhanced by $l_{0.5}$ regularization for sparse damage detection. Zhang et al (2024b) develops a novel Q-learning hybrid evolutionary algorithm (QHEA) that integrates Jaya algorithm, differential evolution, and Q-learning for structural damage identification. This approach enables response reconstruction and damage detection without input excitation, showcasing improved accuracy and robustness in identifying damage locations and severities.

The integration of foundational algorithms with reinforcement learning has shown promising results in optimization and structural damage identification. However, a critical issue in these hybrid models is the reward mechanism – where solutions surpassing previous ones are rewarded, and less optimal outcomes are penalized, directly influencing the optimization performance. In contrast, as an alternative reinforcement learning approach, Learning Automata (LA) offers a distinct approach by selecting appropriate local search strategies without the need for explicit rewards (Hashemi and Meybodi, 2011; Zhang et al, 2020). Instead, feedback is obtained through interaction with the environment, offering positive or negative signals. For instance, in structural damage identification, after executing a chosen strategy, the environment informs the agent whether it is closer to the actual damage location. A closer proximity results in a positive signal (represented as 0), while a move away triggers a negative signal (represented as 1), facilitating an intuitive feedback mechanism without the need for predefined reward policy. On the other hand, the studies abovementioned have seen most of advancements in single-objective optimization techniques, thus, there remains a critical gap in applying improvements to multi-objective optimization algorithms, particularly within our current optimization framework, which demands small yet diverse solutions for structural damage identification. Drawing from the preceding discussion, the research gap is identified as:

1) Single-objective optimization tends to yield a singular solution, potentially confined to a local extremum.
2) A notable challenge within physics-based structural damage identification emerges due to the high-dimensional variables involved and the multimodal objective function peppered with many local optima, underscoring the need for local search strategies.
3) To accurately capture true damage scenarios through a spectrum of solutions, there is a critical need for an approach that navigates this complexity without relying on a predefined reward policy.

These contexts set the stage for presenting our research work, which aims to address these challenges by developing novel strategies for effective and precise damage detection inspired by the success and flexibility of the respective techniques. In this research, we formulate the damage identification problem as a bi-objective optimization using different types of damage information carriers, such as modal information and piezoelectric admittance. The objective functions include Modal Assurance Criterion (MAC) (Greś et al, 2021) to incorporate modal responses in a simplistic problem with numerical simulation, and Root Mean Square Error (RMSE) to incorporate piezoelectric admittance measurements as well as the l_0 norm from a realistic SHM experimental setup. The l_0 norm is employed to minimize the number of identified damage locations, effectively promoting sparsity in the damage indices. This formulation is based on the fact that structural damage typically occurs sparsely, rather than uniformly throughout the structure. Subsequently, the multi-objective particle swarm optimization (MOPSO) as the basic platform for enhancement is selected, owing to its straightforward conceptualization, ease of implementation, computational efficiency, and small number of control parameters. Specifically, we implement time-varying coefficients and mutation-based local search strategies in PSO to form a spectrum of five distinct actions. These actions are facilitated through the design of Learning Automata (LA), where each particle is regarded as an individual LA. Each LA interacts with the environment so it can adaptively select the appropriate action according to its current position. Each time an action is performed, the environment gives feedback to inform the particle whether it is closer or further away from the optima. The particle will then update the probability of the action just performed based on this feedback. The more favorable the action, the higher the probability will be accumulated. It is worth noting that these local search strategies (actions) have their own characteristics and play different roles. The contributions of this study are fourfold:

1) A new inverse analysis approach through multi-objective optimization is synthesized to facilitate structural damage identification utilizing modal information and piezoelectric active interrogation.
2) An enhanced particle swarm optimizer is developed with particles intellectualized through learning automata, enabling a small yet diverse solution set.
3) The particles adaptively move towards optimal solutions based on environmental signals, validated through benchmark case comparison.
4) When implemented to numerical case and experimental data, the proposed approach effectively and accurately identifies structural damage.

The rest of the paper is organized as follows. Section 2 outlines the multi-objective optimization formulation for damage identification by using physical model prediction and experimental measurements. In Section 3, we first outline the necessary details of PSO. We then formulate time-varying coefficients and mutation-based local search strategies realized through LA. In Section 4, the proposed new algorithm is tested on benchmark cases to examine quantitatively the solution diversity and accuracy. Section 5 presents the implementation of the new algorithm to model-based damage identification using modal information and piezoelectric admittance measurements and highlights the performance. Section 6 summarizes the concluding remarks and future research directions.

2 Multi-objective Optimization Setup for Damage Identification

Physical model-based damage identification is an approach to structural health monitoring where the goal is to detect, localize, and quantify damage within a structure through comparison between experimental measurements and model predictions. This process typically involves gathering data from various types of structural responses such as frequencies, mode shapes, accelerations, or piezoelectric impedance, and then utilizing this data to inform a model of the current state of structure. In creating a generalized multi-objective optimization model for damage identification, we integrate both actual/experimental observations and theoretical model predictions. When utilizing modal information, we can establish a multi-objective optimization model with the MAC as the objective function. When employing piezoelectric impedance, we can develop a multi-objective optimization model that targets the Root Mean Square Error (RMSE or l_2 norm) and the l_0 norm as the objective functions. Here l_0 norm aims to enforce the sparsity of the damage representation, reflecting the assumption that actual damage is typically localized and affects a small portion of the structure. Mathematically, the model can be expressed as:

$$\text{Find: } \boldsymbol{\alpha} \in \boldsymbol{E}^n,\ \alpha_l \le \alpha_i \le \alpha_u,\ i = 1, \ldots, n$$

$$\text{Minimize: } f_1 = \left\| Y_{meas} - Y_{pred}(\boldsymbol{\alpha}) \right\|_2 \text{ or } MAC(Y_{meas}, Y_{pred}(\boldsymbol{\alpha})) \tag{1a}$$

$$f_2 = \left\| \boldsymbol{\alpha} \right\|_0 \tag{1b}$$

where $\|\cdot\|_2$ and $\|\cdot\|_0$ denote the l_2 norm and the l_0 norm, respectively. Y is a general variable representing different types of information carries employed for damage identification. $\boldsymbol{\alpha}$ is a damage index vector.

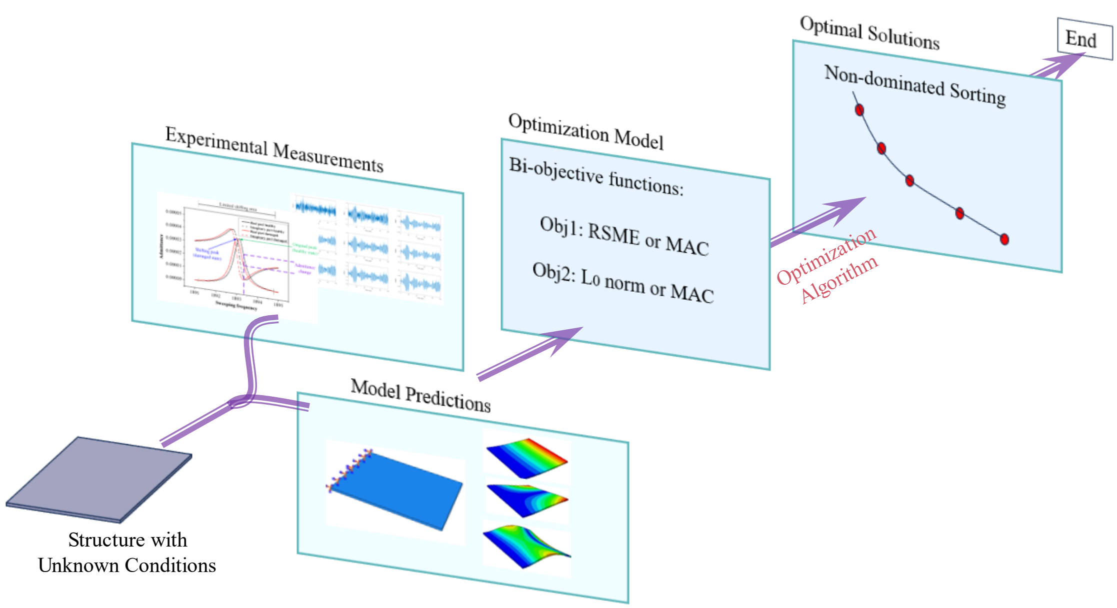


Figure 1. Flowchart of structural damage identification with bi-objective optimization.

The flowchart in Figure 1 depicts the basic steps of a physical model-based approach to damage identification in a structure with an unknown health state, utilizing both experimental data and model predictions. The two branches converge to evaluate differences, which are then fed into a bi-objective optimization process, ultimately yielding a set of solutions characterized by a Pareto front. This formulation of inverse analysis matches with the nature of structural damage identification. It will yield a solution set with multiple solutions that can be used for high level decision making. As stated, the challenge lies in how to handle this high-dimensional, multi-modal optimization problem to find a small yet diverse solution set that can capture the true damage scenario as one of the solutions. Thus, tailoring an algorithm to handle the challenges becomes necessary.

3 Guiding Multi-objective Particle Swarm Optimization through Adaptive Learning Automata

Solving the optimization model formulated for actual damage location and severity is mathematically challenging. On one hand, we pursue solution diversity such that the true damage scenario can be included in the solution set. On the other hand, the size of the solution set will need to be relatively small so it can provide useful insight into high-level decision-making. While a variety of optimizers can potentially be explored, in this research we choose multi-objective particle swarm optimization (MOPSO) as the basic platform for enhancement, owing to its straightforward conceptualization, ease of implementation, computational efficiency, and small number of control parameters. Furthermore, we resort to LA mechanism to develop novel strategies to tackle the issues in MOPSO to realize effective damage identification inverse analysis.

3.1 General framework of particle swarm optimization

Particle swarm optimization is a swarm inspired evolutionary algorithm originally proposed by Eberhart and Kennedy (1995). It is then extended to multi-objective optimization (Coello et al, 2004). PSO is essentially a population-based metaheuristic approach that starts with a randomly generated initial population representing the candidate solutions referred to as particles. The particle moves through a D dimensional space as follows,

$$x_i\left(t+1\right)=x_i\left(t\right)+v_i\left(t+1\right) \tag{2a}$$

$$v_i\left(t+1\right)=zv_i\left(t\right)+c_1r_1\left(p_{\text{best}i}\left(t\right)-x_i\left(t\right)\right)+c_2r_2\left(g_{\text{best}}\left(t\right)-x_i\left(t\right)\right) \tag{2b}$$

where $x_i(t)$ and $v_i(t)$ represent, respectively, the position and velocity of the i-th particle in the t-th iteration. p_{best} is the best position visited by particle i till the t-th iteration, and g_{best} is the global best position amongst all the positions visited by all the particles. z is the inertia weight that reflects the tendency of the particles to maintain their previous speed. c_1 and c_2 are acceleration coefficients. c_1 regulates how much the particle trusts its own experience, and c_2 represents the trend of particles approaching the best position of the entire swarm. r_1 and r_2 are two random numbers chosen from uniform distribution over the range (0, 1). In each iteration, each particle's velocity or position are constrained by the boundary conditions since the particles may exceed the allowed search space and produce invalid solution. The main steps of the PSO algorithm are given as the pseudo-code **Algorithm 1**.

Algorithm 1: Particle swarm optimization framework

(1) Initialize the position and velocity of the particles.
(2) Calculate the fitness value $f(x_i)$ ($i = 1, \cdots, N$) using Equation (1)
(3) For each particle x_i, compare its fitness value $f(x_i)$ with the personal best location that it has experienced. If $f(x_i) < p_{\text{best}}$, update it as the current local best position.
(4) For each particle x_i, compare its personal best fitness value with the global best fitness value. If $f(x_i) < g_{\text{best}}$, update it as current global best position.
(5) Adjust the velocity and position of the population according to Equation (2); perform boundary condition checking.
(6) If the algorithm reaches the maximum number of iterations or the minimum values of the fitness function, stop and output the result; otherwise go to step (2).

3.2 Guidance to search with learning automata

Learning automata (LA) can be regarded as an abstract decision-making unit situated in a stochastic environment that determines the optimal scheme through a set of actions and by frequent interactions with the environment (Hashemi and Meybodi, 2011; Zhang et al, 2020). An automaton contains a finite number of actions, where an action is chosen based on a specific probability distribution and applied to the environment. The environment evaluates the impact of the applied action and then sends back a reinforcement signal to the automaton. The learning algorithm of the automaton utilizes this response to update the existing action probability distributions. By continuing this process, the automaton increases the chance of better actions which generate favorable responses from the environment.

Mathematically, LA are represented by the quadruple (a, β, p, T), where $a = (a_1, \ldots, a_r)$ is a set of actions, $\beta = (\beta_1, ..., \beta_r)$ is a set of input, $p = (p_1, \ldots, p_r)$ is a probability vector regulating the output choices, and T is the learning algorithm used to modify (update) p. The learning algorithm T is a crucial factor affecting the performance of LA. The linear reward-penalty algorithm (L_{rp}) is one of the commonly employed learning algorithms presented in literature (Hashemi and Meybodi, 2011). The learning model T for L_{rp} scheme to update the state probability vector p after receiving the reinforcement signal β is given as follows,

$$p_j(k+1) = \begin{cases} p_j(k) + c\left(1 - p_j(k)\right) & \text{if } i = j \\ p_j(k)(1-c) & \text{if } i \neq j \end{cases} \tag{3a}$$

$$p_j(k+1)=\begin{cases} p_j(k)(1-b) & \text{if } i=j \\ \dfrac{b}{r-1}+p_j(k)(1-b) & \text{if } i\neq j \end{cases} \tag{3b}$$

where r is the number of actions defined in the automaton; parameters c and b represent reward and penalty step length, respectively. Equation (3a) corresponding to the case when $\beta=0$ (favorable response), and Equation (3b) corresponds to the case when $\beta=1$ (unfavorable response). a_i is thus the action chosen at time t as using the distribution $p(t)$. Figure 2 illustrates how LA interacts with the environment.

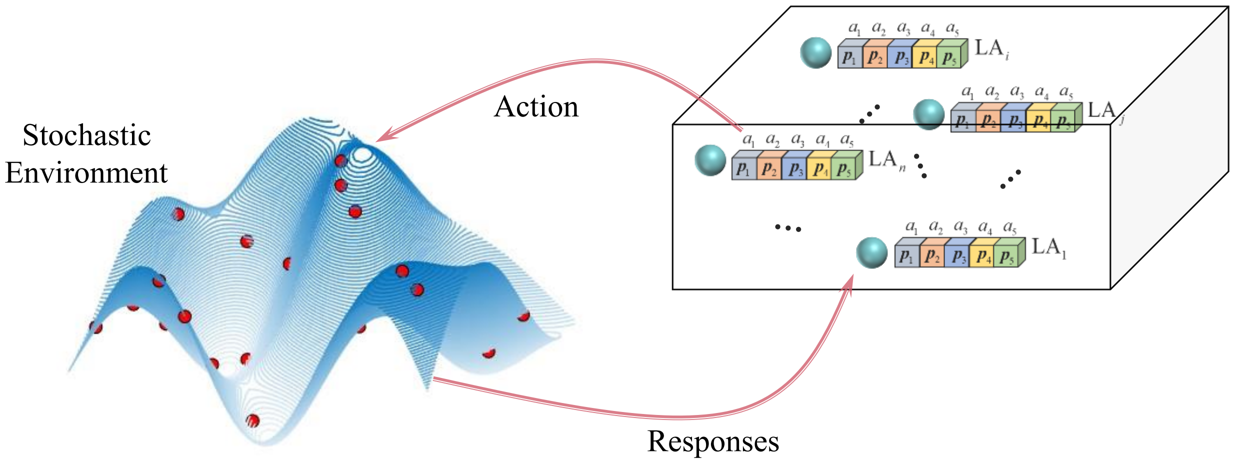


Figure 2. Illustration of learning automata (LA) scheme.

3.3 Learning automata guided MOPSO towards a diverse and small solution set

When solving the damage identification problem formulated in Equations (1a) and (1b), our objective is to obtain a small yet diverse set of solutions that encompasses the true damage scenario. We leverage multi-objective particle swarm optimization (MOPSO) to achieve this goal. The working mechanism of MOPSO, which involves updating particle positions based on individual and collective experiences, offers the potential to incorporate metaheuristics for performance enhancement. Specifically, we synthesize a series of local search strategies using time variation and mutation, treating them as actions. Subsequently, learning automata (LA) is utilized to select the most appropriate action at each iteration, based on feedback from the optimization process. The integration of MOPSO with LA is detailed as follows.

Time varying inertia weight (TVIW): The TVIW was suggested by Shi and Eberhart (1999), which is conducive to early exploration and later exploitation. This can be expressed as $z^t=z_{\max}-t\left(z_{\max}-z_{\min}\right)/t_{\max}$, where $t_{\max}$ is the maximum number of iterations, t is the current iteration, and $z_{\min}$ and $z_{\max}$ are the initial and final values of the inertia weight. Here $z_{\max} = 0.9$ and $z_{\min} = 0.4$ are recommended (Eberhart and Kennedy, 1995).

Time varying acceleration coefficients (TVAC): The TVAC was proposed to enhance the global search ability during early stages as well as the convergence toward the global optima during the later stages (Ratnaweera et al, 2004). This is achieved by modifying two acceleration coefficients c_1 and c_2 in the following manner, $c_1^t = c_1^{\max} - \left(c_1^{\max} - c_1^{\min}\right) t/t_{\max}$, $c_2^t = c_2^{\max} + \left(c_2^{\max} - c_2^{\min}\right) t/t_{\max}$. Here, $c_1^{\min}$, $c_2^{\min}$ are the initial values of c_1, c_2; $c_1^{\max}$ and $c_2^{\max}$ are the final values of the c_1 and c_2, respectively. t is the current iteration number, and $t_{\max}$ is the maximum number of allowable iterations. TVAC changes the cognitive component and the social component by reducing c_1 and increasing c_2 along with the iteration. It can make adjustment between social and cognitive experiences so as to mitigate premature convergence of the algorithm.

Recall Equations (2a) and (2b). Based on these time varying coefficients, we can define two actions as follows,

$$v_i\left(t+1\right) = z^t v_i\left(t\right) + c_1^t r_1\left(p_{\text{best}i}\left(t\right) - x_i\left(t\right)\right) + c_2^{\min} r_2\left(g_{\text{best}}\left(t\right) - x_i\left(t\right)\right) \tag{4a}$$

$$x_i\left(t+1\right) = x_i\left(t\right) + v_i\left(t+1\right) \tag{4b}$$

$$v_i\left(t+1\right) = z^t v_i\left(t\right) + c_1^{\min} r_1\left(p_{\text{best}i}\left(t\right) - x_i\left(t\right)\right) + c_2^t r_2\left(g_{\text{best}}\left(t\right) - x_i\left(t\right)\right) \tag{5a}$$

$$x_i\left(t+1\right) = x_i\left(t\right) + v_i\left(t+1\right) \tag{5b}$$

In both actions, we adopt time varying inertia weights. For Action 1, as shown in Equations (4a) and (4b), we use time varying acceleration coefficient c_1 and keep c_2 as constant. This can help particles to explore more space at the beginning of the optimization process. On the other hand, Action 2, as shown in Equations (5a) and (5b), takes c_1 as constant and uses a time varying acceleration c_2. Through this, the search process can achieve a convergence at the end of the optimization. The new position, which is the potential solution, will be updated using any action. Then the fitness values can be calculated and compared to the values from the old solution through dominant sorting algorithm. Positive feedback will be given as 0 if the new solution dominates the old one; otherwise, a negative feedback will be given as 1 for the action selected at current iteration.

To help particles jump out of potential local extremes, an ***Elitist-based perturbation*** (EBP) strategy, Action 3, is adopted to perform perturbation on the personal best of the particle and provide extra diversity (Lim and Isa, 2013). To trigger the EBP strategy, we introduce a variable called *RT* that determines the number of times the EBP strategy is repeated. In the EBP strategy, a randomly selected dimension in the personal best of a particle is perturbed using following logic,

$$tempP_{\text{best}i,d} = \begin{cases} p_{\text{best}i,d} + r_d \cdot \left(\alpha_{\max,d} - \alpha_{\min,d}\right) & \text{if rand} > 0.5 \\ p_{\text{best}i,d} - r_d \cdot \left(\alpha_{\max,d} - \alpha_{\min,d}\right) & \text{if rand} \leq 0.5 \end{cases} \tag{6}$$

Here $tempP_{\text{best}i,d}$ is an intermediate variable that temporarily represents the personal best, $p_{\text{best}i,d}$ denotes 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 $N \sim \left(\mu, R^2\right)$ with $\mu = 0$ and standard deviation of *R* where $R = R_{\max} - t(R - R)/t_{\max}$. $R_{\max} = 1$ and $R_{\min} = 0.1$, as suggested by Lim and Isa (2013), are the maximum and minimum perturbation ranges, respectively. *t* is the current number of iterations, and $t_{\max}$ is the maximum iteration of the algorithm. $\alpha_{\min,d}$ and $\alpha_{\max,d}$ are the lower and upper bounds of the *d*-th dimension. The personal best will be replaced by the temporary personal best $tempP_{\text{best}i,d}$, if the latter dominates the former and the reinforcement signal will be given as 0 to indicate a favorable action. In other words, the personal best finds a new position which is closer to global solution or closer to true damage scenario in structural damage identification through EBP; otherwise, the personal best keeps unchanged, and the signal is 1 to show an unfavorable action, i.e., the action achieves nothing but costs computation time, so the penalty is necessary.

Another emphasis called ***Mutation*** (Action 4) aims at assisting the global best in examining its surroundings and determining whether it can break free from its current position and discover a more advantageous one. We use *L* to represent the global best, and have

$$tempL_{\text{best},d} = L_{\text{best},d} + norm(\mu, R^2) \tag{7}$$

where $tempL_{\text{best}i,d}$ is an intermediate variable that temporarily represents the global best. As shown above, $L_{\text{best},d}$ denotes the *d*-th dimension of the global best. A random perturbation is generated following the same normal distribution in strategy EBP. The decay standard deviation is used to attain a significant jump at the beginning and a smaller one at the end of the process. At this time, if the temporary global best dominates the global best, then it will replace the old one. The reinforcement signal is then given as 0 to show that ***Mutation*** is favorable. Otherwise, the signal is given as 1.

During the ***Fine-Tuning*** process (Action 5), the focus is on every dimension of the personal best through its velocity updating, rather than only one randomly selected dimension. This strategy aims to encourage particles to move more in the direction that enhances the solution and, conversely, to slow down in the direction that impedes solution improvement. To initiate the action, tuning is repeated *PT* times for each dimension. However, it is crucial to avoid setting *PT* to a high value because optimization deals with high-dimensional problems, and an excessively large *PT* value can prolong the runtime significantly. Here we set *PT* as 10. Let $V_{i,d}$ denote the velocity of the *d*-th dimension of the personal best of the *i*-th population in fine tuning. We employ the following in this research (Ji et al, 2007; Samma et al, 2016),

$$V_{i,d} = \frac{\varphi}{t^{\phi}} r + 2B_{i,d} \tag{8}$$

where φ is the acceleration factor and set as 2, ϕ is the descent parameter that controls the decay of the velocity and is set as 20, r is a uniformly distributed random number within [−0.5,0.5], and t is the current iteration number. Moreover, we let

$$B_{i,d} = \begin{cases} 2V_{i,d}, & if\ x_{\text{new}} \prec x_{\text{old}} \\ 0.5B_{i,d}, & \text{otherwise} \end{cases} \tag{9}$$

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

$$tempP_{\text{best}i,d} = \begin{cases} p_{\text{best}i,d} + V_{i,d}, & if\ x_{\text{new}} \prec x_{\text{old}} \\ p_{\text{best}i,d}, & \text{otherwise} \end{cases} \tag{10}$$

The updated temporary best $tempP_{\text{best}i,d}$ will take replace of the personal best if it dominates the personal best. Otherwise, the original personal best remains unchanged. Subsequently, the positive signal 0 or negative signal 1 will be applied accordingly. The integration of the abovementioned strategies through LA into MOPSO is outlined as pseudo code shown in **Algorithm 2**.

Algorithm 2: Learning Automata guided MOPSO

(1) Initialization: Randomly generate the positions and velocities of all particles, initialize the probability of selecting the global search strategy p as $p = 1/o(a)$, where p represents the probability that the agent selects the global search strategy and $o(a)$ is the number of action. Randomly choose an action, Equations (4) -(8).
(2) Position and Velocity Updating:
   - *2*.1 Update the velocity of current learning automaton using the selected action
   - *2*.2 Update the position of current learning automaton using the selected action
   - *2*.3 Deal with the bounds
   - *2*.4 Calculate the fitness values
(3) Update the personal best using non-dominated sorting algorithms
(4) Update the global best using non-dominated sorting algorithms
(5) Delete the solutions if the number of solutions in Pareto front is reached.
(6) Update Probability: If updated solution dominates the old solution, update the probability using Equation (3a); otherwise, update the probability using Equation (3b)
(7) Terminal condition check: If any of pre-defined termination criteria is not satisfied, repeat from step 2. Otherwise, the algorithm terminates.

3.4 Summary of the proposed optimization algorithm

The proposed LA guided MOPSO algorithm, referred to as the LAMOPSO hereafter, is illustrated through the flowchart shown in Figure 3, in which the steps from top to bottom correspond to the steps in Algorithm 2. After initialization, each population is regarded as a learning automaton (LA). Each LA has 5 actions, as described in the previous subsection. This LAMOPSO utilizes the ability of LA to learn and adaptively select a suitable strategy for the swarm in various search stages based on feedback received.

Initially, LA randomly selects an action as all actions have an equal probability. As the search progresses, LA learns from feedback in each iteration and updates the selection probabilities of candidate actions. In later stages, after interacting with the environment, LA identifies the actions that yield positive results and accumulates the corresponding probabilities. It is noteworthy that the update of the selection probabilities of candidate strategies is not predetermined but learned from the evolution process via LA, which promotes the intelligence of the swarm.

The LAs are independent individuals since they have their own actions and corresponding probabilities. The advantage is that they can decide their next moves based on the current landscape. Meanwhile, they work in a collaborative way, as they update the same optimal solution repository. In order to obtain a small yet diverse set of solutions that accurately represent the true damage scenario, we have devised five search methods that rely on coefficient variation and mutation. These techniques augment the search process, thereby facilitating the acquisition of a diverse set of solutions. Furthermore, we have implemented Learning Automata to help the particles intelligently choose the appropriate strategy during each iteration.

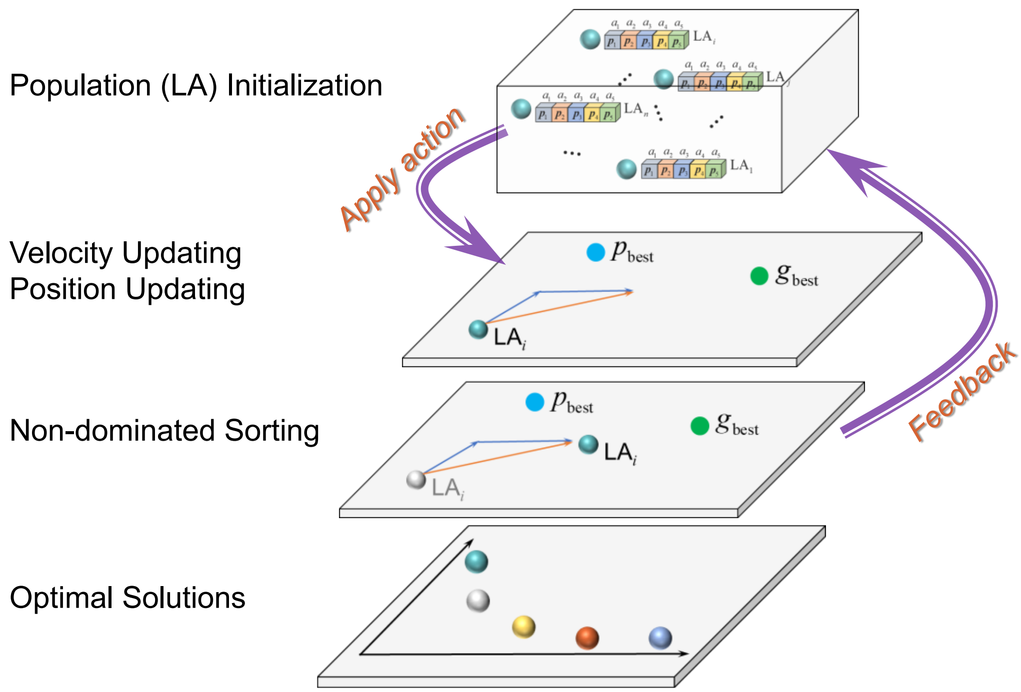


Figure 3. Flowchart of the proposed LAMOPSO algorithm.

4 Demonstration and Validation through Case Investigations

Our goal is to synthesize a new multi-objective optimization solver that can be effectively implemented to structural damage identification to tackle the peculiar challenges. Prior to examining the performance in structural damage identification, we first conduct case demonstration of the LAMOPSO developed in the preceding section on benchmark cases. This leads to, first and foremost, algorithmic implementation insights, and provides performance comparison with respect to representative optimization solvers to showcase the efficacy of the new approach.

4.1 Benchmark test functions

The topic of multi-objective optimization is quite broad, and a number of methods have been synthesized in recent years to tackle various challenges. In this section, without loss of generality, we conduct the performance comparison employing the following test functions: ZDT (Zitzler er al, 2000), CEC09 or UF (Zhang et al, 2008) and IMOP (Tian et al, 2019). The proposed LAMOPSO is compared with multi-objective optimization solvers including NMPSO (Lin et al, 2018), MMOPSO (Lin et al, 2015), GPSOM (Noel, 2012), MO_Ring_PSO_SCD (Yue et al, 2018), MOEADDQN (Tian et al, 2023), CMOQLMT (Ming et al, 2023), MOCS (Du and Trung, 2023).

Table 1. Benchmark test functions.

Problems	*M*	*D*	Population size	Descriptions
UF1	2	30	50	Nonlinear decision space
UF2	2	30	50	Nonlinear decision space
UF3	2	30	50	Convex front
UF4	2	30	50	Concave front
UF5	2	30	50	Discrete points on a linear hyperplane
ZDT1	2	30	50	Convex front
ZDT2	2	30	50	Concave front
ZDT3	2	30	50	Discontinuous front
ZDT4	2	30	50	Convex front, many local optimal
ZDT6	2	30	50	Concave front, non-uniform distribution
IMOP1	2	15	50	Convex front with sharp tail
IMOP2	2	15	50	Concave front with sharp tail
IMOP3	2	15	50	Discontinuous front

The algorithms selected for comparison exhibit a high degree of competitiveness. They are either variants of the PSO algorithm or variants of evolutionary algorithms and reinforcement learning techniques, such as Deep Q-Networks and Q-Learning, which all are the most directly related to current research. These algorithms have demonstrated promising performance in the domain of multi-objective optimization. Meanwhile, the test functions selected are frequently used in the past to evaluate the performance of metaheuristics. Especially, the IMOP test functions are designed in recent years, which have sharp Pareto front, thus posing challenges to the metaheuristics to converge to the true Pareto front. The test function details are listed in Table 1. These functions are employed to evaluate the convergence of the algorithms and the diversity of solutions generated by the algorithms.

4.2 Insights into parametric settings and evaluation metrics

In the benchmark testing, *D* is the dimension of each test problem and *M* is the number of objectives. Here, *M* is set as 2 for all testing instances. *D* is 30 for test functions ZDTs and UFs, and 15 for IMOPs. The true Pareto front of the IMOP test functions is quite sharp, which poses significant challenge. Thus, we use 15 for the IMOPs to mitigate the difficulty for all the algorithms to be evaluated. The variables in the IMOPs can be written as $\mathbf{x}=\left(x_1,\ldots,x_L,x_{L+1},\ldots,x_D\right)\in\left[0,1\right]$. We apply three objective functions for the IMOPs, in which $y_1=\left(1/L\sum_{l=1}^{L}x_l\right)^{a_1}$, $g=\sum_{l=L+1}^{D}\left(x_l-0.5\right)^2$. Here, for the IMOPs, the two parameters involved are set as $a_1=0.05$ to control the mathematical difficulty in diversity preservation and $L=5$ to control the length of variables used for calculations of y_1 and *g*. The specific expressions for all test functions are included in the appendix for reference. The total number of iterations for all test problems is set as 40,000 after convergence analysis based on two indicators as explained below.

In this research, we utilize the hypervolume (HV) and the inverted generational distance (IGD) to quantify the performance of the algorithms. The performance comparison is based on the Pareto set that is a set of solutions realizing the optimal trade-offs between the optimization objectives. The HV indicator defined below measures the convergence as well as the diversity (Friedrich et al, 2009),

$$HV\left(X\right)=\mathrm{VOL}\left(\bigcup_{x\in X}(x,P)\right) \tag{11}$$

where *X* is a solution set obtained by one algorithm, $\mathrm{VOL}(\cdot)$ is the Lebesgue measure, and *P* is a set of reference points. The calculation of *HV* requires the normalized objective function values. Indeed, *HV* stands for the percentage covered by the Pareto front of the cuboid defined by the reference point *P* and the original point *x*. Here the reference point is set to be 1.1 times the upper bound of the normalized *PF*. The hypervolume metric can also assess both the convergence and diversity of the dominant solution set. A larger *HV* value indicates better diversity of solutions.

IGD is an indicator measuring the degree of convergence by computing the average of the minimum distance of points in the true Pareto to points in Pareto front obtained (Ishibuchi et al, 2018), i.e.,

$$IGD\left(PF,PF^*\right)=\frac{\sum_{\mathbf{f}^*\in PF^*,\ i=1}^{|PF^*|}\sqrt{\min_{\mathbf{f}\in PF}\left(\sum_{m=1}^{M}(f_m^{*i}-f_m)^2\right)}}{|PF^*|} \tag{12}$$

where *M* is the number of objectives, PF^* is the true Pareto front for a specific test function; $|PF^*|$ is the cardinality of PF^* and *PF* is the Pareto front obtained by one algorithm, f_m^* and f_m are the *m*-th objective values in true Pareto set and Pareto set calculated by one algorithm, respectively. The essence of Equation (12) is to calculate the Euclidean distance between the points on the true Pareto front and the points on the

Pareto front obtained. Therefore, a smaller value of *IGD* means that the two fronts are closer to each other. In other words, a smaller value of *IGD* indicates a better convergence and completeness of the *PF* obtained.

4.3 Discussion on benchmark test case results

All algorithms are executed independently 30 times for each test problem on platform PlatEMO (Tian et al, 2017). The means and variances of the *HV* and *IGD* results obtained from 30 independent runs on UF, ZDT and IMPO are listed in Tables 2 and 3, respectively. The best mean values of the *HV* index in each test function are highlighted with shade in the table. In each test case, the first-row values are the means, and the second-row parenthesized values are the variances.

Table 2. *HV* results from 30 executions with two problem objectives.

Problems	Alg1	Alg2	Alg3	Alg 4	Alg 5	Alg 6	Alg 7	Alg 8
UF1	5.864e-1	6.082e-1	5.195e-1	4.841e-1	6.408e-1	6.630e-1	5.974e-1	5.872e-1
	(1.02e-2)	(3.46e-2)	(6.93e-2)	(8.00e-3)	(3.83e-2)	(1.32e-2)	(1.28e-2)	(1.12e-2)
UF2	6.113e-1	6.716e-1	6.426e-1	3.768e-1	6.897e-1	6.857e-1	7.118e-1	7.122e-1
	(2.17e-2)	(3.42e-3)	(1.12e-2)	(1.24e-2)	(5.67e-3)	(4.00e-3)	(1.33e-2)	(1.57e-2)
UF3	5.283e-1	4.277e-1	3.090e-1	1.551e-1	5.257e-1	5.691e-1	2.630e-1	4.436e-1
	(7.02e-2)	(5.05e-2)	(3.35e-2)	(4.10e-3)	(1.03e-1)	(2.76e-2)	(1.79e-2)	(2.26e-2)
UF4	3.351e-1	3.686e-1	1.447e-1	2.516e-1	3.629e-1	3.795e-1	3.407e-1	3.880e-1
	(2.08e-2)	(5.64e-3)	(3.17e-2)	(3.04e-3)	(6.88e-3)	(2.00e-3)	(1.83e-2)	(2.49e-3)
UF5	4.006e-2	4.082e-2	0.000e+0	0.000e+0	1.340e-1	2.633e-1	2.160e-1	2.762e-1
	(5.06e-2)	(4.40e-2)	(0.00e+0)	(0.00e+0)	(7.21e-2)	(4.80e-2)	(8.17e-2)	(6.45e-2)
ZDT1	6.920e-1	7.133e-1	6.976e-1	2.308e-1	7.158e-1	7.154e-1	7.079e-1	7.163e-1
	(7.45e-3)	(6.24e-4)	(2.31e-3)	(1.29e-2)	(9.74e-5)	(3.64e-4)	(1.43e-2)	(6.52e-4)
ZDT2	4.330e-1	4.384e-1	1.736e-1	0.000e+0	4.408e-1	4.398e-1	3.713e-1	4.418e-1
	(2.66e-3)	(5.20e-4)	(0.00e+0)	(0.00e+0)	(8.33e-5)	(3.85e-4)	(1.55e-1)	(9.55e-3)
ZDT3	5.670e-1	5.972e-1	5.312e-1	3.940e-1	5.947e-1	5.977e-1	8.300e-1	8.399e-1
	(3.33e-4)	(2.25e-4)	(3.63e-2)	(2.20e-2)	(2.86e-5)	(1.54e-4)	(9.39e-3)	(1.47e-2)
ZDT4	0.000e+0	5.975e-3	0.000e+0	0.000e+0	1.314e-1	0.000e+0	0.000e+0	7.176e-1
	(0.00e+0)	(2.22e-2)	(0.00e+0)	(0.00e+0)	(1.86e-1)	(0.00e+0)	(0.00e+0)	(1.72e-3)
ZDT6	3.839e-1	3.832e-1	1.180e-1	0.000e+0	3.857e-1	3.776e-1	4.091e-1	4.247e-1
	(7.71e-4)	(5.46e-4)	(3.49e-2)	(0.00e+0)	(2.59e-4)	(8.16e-3)	(1.20e-2)	(1.21e-4)
IMOP1	9.568e-1	9.856e-1	8.126e-1	2.982e-1	9.823e-1	9.863e-1	9.862e-1	9.880e-1
	(1.78e-2)	(6.26e-4)	(8.24e-2)	(2.17e-2)	(8.01e-5)	(2.20e-4)	(2.82e-3)	(3.71e-4)
IMOP2	2.186e-1	2.294e-1	1.326e-1	8.442e-2	2.271e-1	2.296e-1	2.018e-1	1.010e-1
	(4.79e-3)	(3.49e-4)	(4.19e-2)	(7.81e-4)	(1.52e-3)	(2.77e-4)	(5.16e-2)	(2.03e-4)
IMOP3	5.912e-1	6.559e-1	2.319e-1	1.355e-1	6.540e-1	6.492e-1	3.790e-1	6.561e-1
	(3.19e-2)	(5.94e-4)	(4.43e-2)	(6.29e-3)	(3.28e-4)	(1.08e-2)	(1.74e-2)	(3.43e-4)

** Algorithms 1 to 8 in Table 2 are: NMPSO, MMOPSO, GPSOM, MO_Ring_PSO_SCD, MOEADDQN, CMOQLMT, MOCS and LAMOPSO.

As shown in Table 2, in terms of *HV* indicator, the proposed LAMOPSO outperforms the MOPSO variants such as NMPSO, MMOPSO, GPSOM, MO_Ring_PSO_SCD (the first 4 algorithms) and multi-objective cuckoo search (MOCS) (the 7th column) quite significantly. Especially, the variants GPSOM and MO_Ring_PSO_SCD have many *HV* of zeros on some test functions such as UF5 and ZDT2, 4, 6, indicating that the algorithm is trapped in local optima and unable to achieve diversity in the solution set. Moreover, the LAMOPSO outperforms all other algorithms on the UF and ZDT test sets. Interestingly, the two other algorithms, MOEADDQN and COMQLMT, are variants that incorporate reinforcement learning techniques. However, when pitted against our proposed LAMOPSO algorithm, they exhibit inferior overall performance. Across the majority of the test functions, these algorithms fail to match the capabilities of our LAMOPSO algorithm. The LAMOPSO leads to the largest mean *HV* values and entertains mostly smaller variances indicating more consistent results.

For the IMOP test functions, the LAMOPSO algorithm performs better on IMOP1. On the IMOP2 test function, CMOQLMT demonstrates superior performance, with MMOPSO emerging as its closest competitor, exhibiting only a marginally inferior result. For the IMOP3 test function, LAMOPSO takes a slight lead, while MMOPSO, MOEADDQN, and CMOQLMT algorithms put up a formidable competition, trailing closely behind. As mentioned earlier, the IMOP test functions have sharp Pareto fronts. IMOP2 has a concave front, and IMOP3 has a discontinuous front. The variants incorporating reinforcement learning techniques exhibit a general superiority compared to the variants of traditional algorithms. This observation suggests that reinforcement learning holds significant potential for enhancing and improving optimization algorithms. Overall, the proposed LAMOPSO yields the best performance of *HV* in 10 out of 13 test cases, which illustrates the outstanding performance in diversity and convergence.

Table 3 shows the *IGD* results comparison. Recall that smaller *IGD* values correspond to better convergence and completeness of the Pareto front obtained. It can be observed that the LAMOPSO outperforms other algorithms on most test sets, except for the UF1, 3, 4, IMOP2, and IMOP3 test functions which are won by the CMOQLMT and MMOPSO. It is noteworthy that the CMOQLMT algorithm, a variant that combines evolutionary algorithms with Q-learning, exhibits considerable competitiveness in terms of the IGD metric. This observation further reinforces the potential of reinforcement learning techniques in enhancing the performance of optimization algorithms. The learning automata employed in this work, being a form of reinforcement learning, is utilized to improve upon the traditional MOPSO algorithm, thereby manifesting this inherent potential. Overall, the LAMOPSO yields the best IGD results in 8 out of 13 test cases, and the results are consistent with generally small variances. This indicates that the LAMOPSO can converge well to the true Pareto front.

Table 3. *IGD* results from 30 executions with two problem objectives.

Problems	Alg1	Alg 2	Alg 3	Alg 4	Alg 5	Alg 6	Alg 7	Alg 8
UF1	1.132e-1 (7.42e-3)	7.836e-2 (2.28e-2)	1.307e-1 (4.64e-2)	1.626e-1 (7.95e-3)	5.987e-2 (3.41e-2)	4.255e-2 (1.04e-2)	8.220e-2 (1.25e-3)	9.121e-2 (9.97e-4)
UF2	8.232e-2 (1.46e-2)	3.986e-2 (2.98e-3)	6.586e-2 (8.54e-3)	2.581e-1 (1.34e-2)	2.753e-2 (6.19e-3)	2.927e-2 (3.81e-3)	1.144e-2 (4.17e-4)	9.060e-3 (6.11e-4)
UF3	1.451e-1 (6.44e-2)	2.127e-1 (3.49e-2)	3.124e-1 (3.45e-2)	5.603e-1 (7.23e-3)	1.355e-1 (7.37e-2)	1.131e-1 (2.86e-2)	4.033e-1 (1.01e-3)	2.338e-1 (2.12e-2)
UF4	7.956e-2 (1.59e-2)	5.511e-2 (4.61e-3)	2.125e-1 (3.21e-2)	1.323e-1 (2.33e-3)	6.069e-2 (5.67e-3)	4.735e-2 (1.43e-3)	7.674e-2 (2.21e-3)	5.210e-2 (2.71e-2)
UF5	6.635e-1 (1.48e-1)	6.824e-1 (1.65e-1)	1.936e+0 (3.05e-1)	2.244e+0 (1.39e-1)	4.876e-1 (1.13e-1)	2.422e-1 (4.55e-2)	2.616e-1 (7.57e-2)	2.294e-1 (1.52e-1)
ZDT1	2.714e-2 (7.00e-3)	1.015e-2 (6.30e-4)	2.511e-2 (1.06e-3)	4.312e-1 (1.96e-2)	7.854e-3 (8.48e-6)	8.120e-3 (1.95e-4)	1.005e-2 (7.29e-4)	5.950e-3 (9.97e-4)
ZDT2	2.223e-2 (4.29e-3)	1.064e-2 (5.44e-4)	3.549e-1 (1.69e-16)	9.335e-1 (1.11e-1)	7.701e-3 (6.03e-6)	8.137e-3 (1.58e-4)	1.303e-1 (2.53e-1)	6.120e-3 (6.11e-4)
ZDT3	1.004e-1 (1.03e-3)	1.176e-2 (5.27e-4)	1.015e-1 (3.00e-2)	3.282e-1 (1.56e-2)	2.131e-2 (3.03e-5)	1.009e-2 (2.98e-4)	1.293e-2 1.81e-3	7.100e-3 (2.12e-2)
ZDT4	9.463e+0 (5.50e+0)	1.834e+0 (1.03e+0)	2.573e+2 (1.29e+2)	2.744e+2 (1.11e+1)	7.894e-1 (4.41e-1)	1.994e+0 (6.75e-1)	1.140e+1 (2.04e+0)	5.210e-2 (2.71e-2)
ZDT6	8.529e-3 (9.55e-4)	8.938e-3 (5.64e-4)	5.697e-1 (1.25e-1)	5.066e+0 (6.10e-2)	6.294e-3 (5.45e-5)	1.067e-2 (4.91e-3)	8.280e-3 (4.32e-4)	4.986e-3 (1.52e-1)
IMOP1	3.190e-1 (4.74e-2)	1.230e-2 (5.59e-4)	1.653e-1 (2.30e-2)	6.070e-1 (1.74e-2)	1.233e-1 (1.43e-3)	1.149e-2 (7.80e-4)	1.313e-2 (8.06e-4)	1.136e-2 (5.22e-4)
IMOP2	9.503e-2 (2.69e-2)	1.591e-2 (2.84e-3)	4.900e-1 (1.32e-1)	6.579e-1 (3.05e-2)	3.874e-2 (1.87e-2)	2.095e-2 (5.41e-3)	1.670e-1 (3.26e-1)	2.066e-1 (6.27e-3)
IMOP3	9.718e-2 (2.02e-2)	9.915e-3 (1.33e-3)	3.004e-1 (2.47e-2)	5.847e-1 (3.11e-2)	1.591e-2 (8.99e-4)	1.880e-2 (1.19e-2)	1.018e-2 (1.14e-3)	9.530e-2 (4.23e-4)

** Algorithms 1 to 8 in Table 2 are: NMPSO, MMOPSO, GPSOM, MO_Ring_PSO_SCD, MOEADDQN, CMOQLMT, MOCS and LAMOPSO.

5 Applying the LAMOPSO to Structural Damage Identification Practice

As outlined in Section 2, the physical model-based damage identification can be cast into a multi-objective optimization problem utilizing different types of damage information carries. A unique challenge, however, lies in how to handle such a high-dimensional, multi-modal optimization problem to find a small yet diverse solution set that can capture the true damage scenario as one of the solutions. In this section, we conduct case studies to examine the proposed algorithm improvements using simulation modal responses and experimentally acquired piezoelectric admittance measurements.

5.1 Numerical case study

In this first case study, we apply the proposed algorithm to a simplistic SHM case study based on numerical simulation. This allows interested readers to quickly examine the algorithm details and gain insights.

5.1.1 Damage identification with modal information

Vibration-based SHM relies on modal information obtained from dynamic analysis. By comparing measured and predicted natural frequencies and mode shapes, the locations and severities of damages can be identified. However, in practical engineering applications, the number of unknowns typically exceeds the number of useful measurements, rendering the structural identification an underdetermined problem. As stated in Section 2, in this research we adopt a bi-objective optimization approach to address the underdetermination. Here a correlation coefficient termed the modal assurance criterion (MAC) is utilized to compare two frequency change vectors, i.e., measured frequency change $\boldsymbol{\Delta\lambda}$ and predicted frequency change $\delta\boldsymbol{\lambda}$, as expressed below (Cao et al, 2018),

$$\mathrm{MAC}(\boldsymbol{\lambda},\boldsymbol{\alpha}) = \frac{\left\langle \Delta\boldsymbol{\lambda},\, \delta\boldsymbol{\lambda}(\boldsymbol{\alpha})\right\rangle^2}{\left\langle \Delta\boldsymbol{\lambda},\, \Delta\boldsymbol{\lambda}\right\rangle \cdot \left\langle \delta\boldsymbol{\lambda}(\boldsymbol{\alpha}),\, \delta\boldsymbol{\lambda}(\boldsymbol{\alpha})\right\rangle} \tag{13}$$

where $\langle *, *\rangle$ calculates the inner product of two vectors, and $\boldsymbol{\alpha}$ is the damage index vector. $\mathrm{MAC}(\boldsymbol{\lambda},\boldsymbol{\alpha}) \in [0,1]$ captures the similarity between $\boldsymbol{\Delta\lambda}$ and $\delta\boldsymbol{\lambda}$. The value of MAC equals to one means that two vectors compared are identical to each other in terms of direction. On the other hand, among the modal parameters of a structural system, the mode shape is obviously the only location-related parameter. Mode shape is more sensitive to local damage than modal frequency. It can provide more information about structural damage if we combine frequency change and mode shapes together. Thus, in order to take more local damage information into consideration, for the *j*-th mode shape, we can compare the measured change and predicted change using MAC in a similar manner,

$$\mathrm{MAC}(\phi,\boldsymbol{\alpha}) = \frac{\left\langle \{\Delta\phi_j\},\, \{\delta\phi_j(\boldsymbol{\alpha})\}\right\rangle^2}{\left\langle \{\Delta\phi_j\},\, \{\Delta\phi_j\}\right\rangle \cdot \left\langle \{\delta\phi_j(\boldsymbol{\alpha})\},\, \{\delta\phi_j(\boldsymbol{\alpha})\}\right\rangle} \tag{14}$$

Since both MACs are functions of $\boldsymbol{\alpha}$ and the measured data $\Delta\boldsymbol{\lambda}$ and $\{\Delta\phi_j\}$ are known, a multi-objective minimization problem can then be formulated as Equation (15). Here in the first objective function, we give the same weight for both MAC calculations from frequency and mode shape. The second objective function as mentioned is to regulate the undetermined problem since the measurement is usually limited.

$$\text{Find: } \boldsymbol{\alpha} \in \boldsymbol{E}^n,\ \alpha_l \le \alpha_i \le \alpha_u,\ i = 1,\ldots,n \tag{15a}$$

$$\text{Minimize: } f_1 = -0.5 \cdot \mathrm{MAC}(\boldsymbol{\lambda}) - 0.5 \cdot \mathrm{MAC}(\phi) \text{ and } f_2 = \|\boldsymbol{\alpha}\|_0 \tag{15b}$$

5.1.2 Damage identification results

We first conduct a test on a 30-element cantilever beam with a length of 10 mm per element, as shown in Figure 4. The beam has a Young's modulus of 69 GPa, a width of 50 mm, and a height of 20 mm for the cross-section. Without loss of generality, we utilize the first 6 natural frequencies and the 2nd mode shape as the simulated measurement data, obtained directly from the model in the numerical experiment. Two damage scenarios were considered: Case 1 with a single damage location at element 4 with a 24% stiffness reduction, and Case 2 with two damage locations at elements 4 and 15, with severities of 24% and 15%, respectively. It is noteworthy that the algorithm does not have prior knowledge of the number, locations, or severities of the damages. In this case study with 30 variables, we have set the maximum number of evaluations to 40,000, aligning with the standard used in benchmark test evaluations. As a result of the multi-objective optimization approach, we obtained a set of optimal solutions representing the trade-off between the objectives.

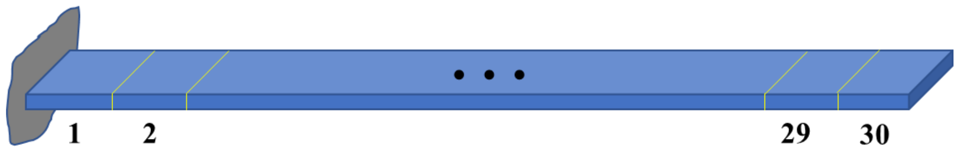


Figure 4. Cantilever beam structure with 30 elements.

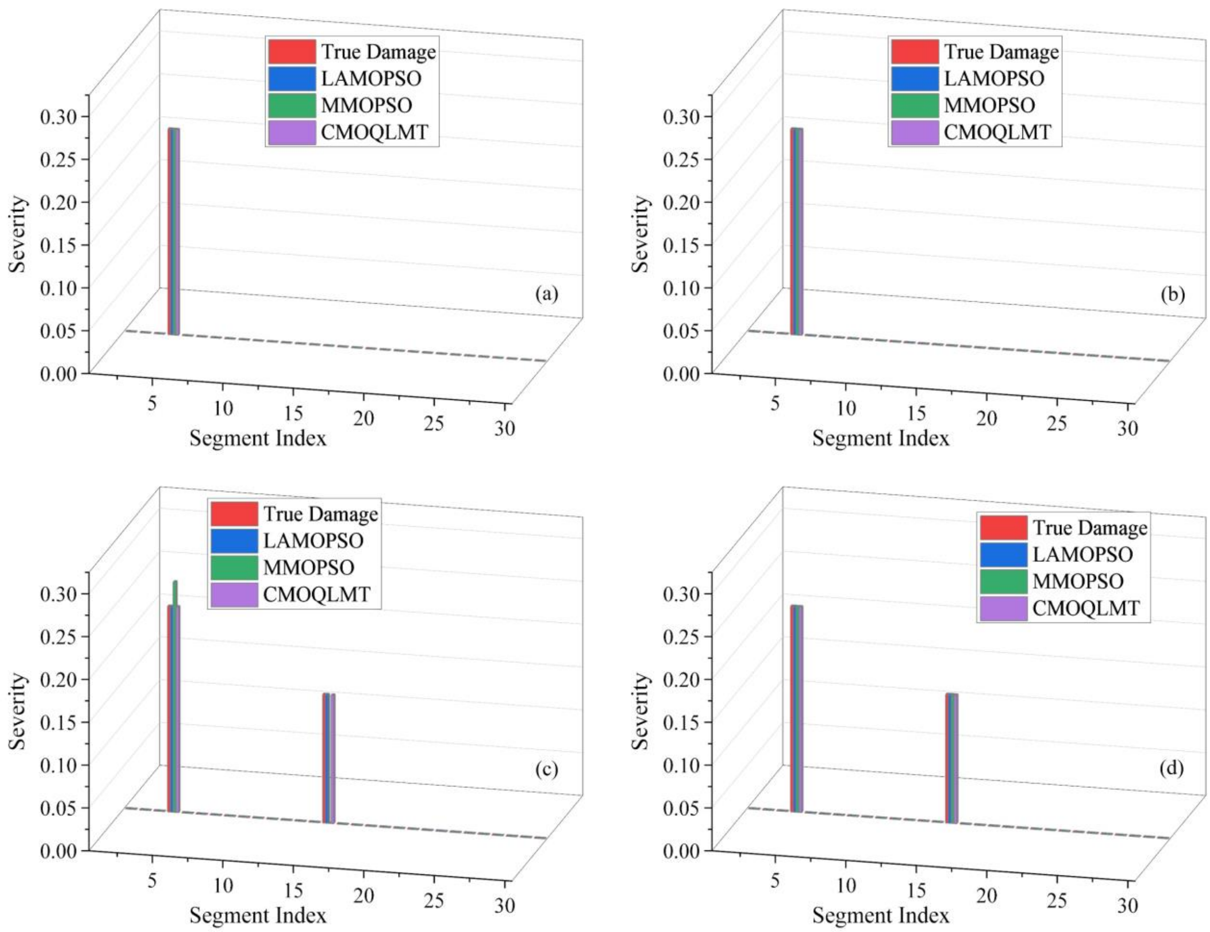


Figure 5. Identified results for cantilever beam structure for Case 1 and Case 2.

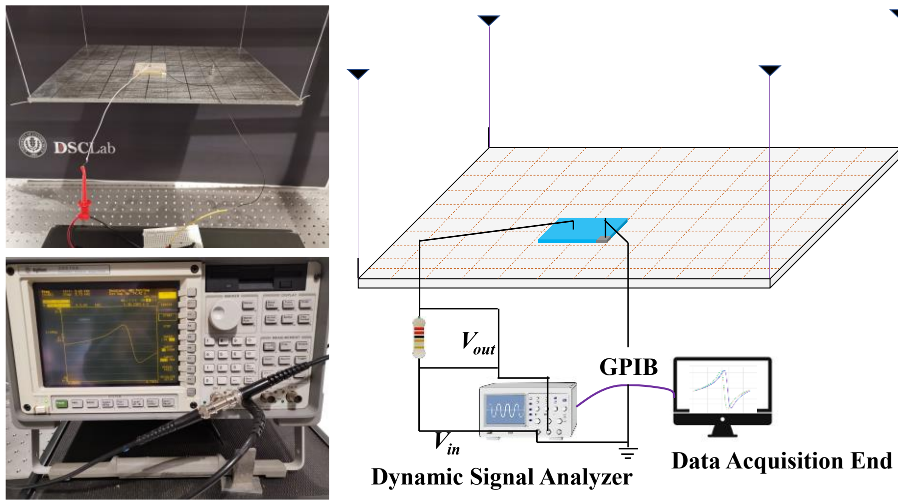
Figure 5 presents the damage identification results obtained by the proposed LAMOPSO algorithm. The top row displays two solutions corresponding to Case 1 (single damage location), while the bottom row shows two solutions for Case 2 (multiple damage locations). In previous benchmark tests, MMOPSO and CMOQLMT exhibited considerable competitiveness on certain test functions. Consequently, their performance was also evaluated on this engineering problem for comparison. As evident from the figure, two direct observations can be made: 1) a small solution set is obtained, which aligns with the expectation for underdetermined problem; 2) the obtained solutions show good sparsity due to the second objective function defined. Specifically, LAMOPSO and CMOQLMT algorithms accurately identify the true damage scenarios, regardless of the number of damage locations or varying damage severities. For MMOPSO algorithm, it behaves well in the case where the single damage happens, as shown in Figure 5(a) and (b). While, in the second case with multiple damage, its first solution misses one true damage location, as shown in Figure 5(c). Overall, all the optimizers including the proposed LAMOPSO algorithm show good performance on this damage identification for beam structure.

5.2 Experimental case study

In this section, we present the application of the proposed algorithm to a realistic SHM experimental setup built upon piezoelectric active interrogation. The piezoelectric active interrogation typically employs a single piezoelectric transducer as actuator and sensor concurrently to collect the impedance or admittance information in near real-time (Fan et al, 2019; Ezzat et al, 2020). The impedance or admittance information can be collected in high-frequency range with smaller wavelengths, paving a way for high-accuracy damage identification when combined with a high-fidelity finite element model. Such high-fidelity finite element model, however, usually features high dimensionality, posing significant challenge to inverse identification.

5.2.1 Experimental setup and physical modeling

Here we employ piezoelectric admittance-based active interrogation for structural damage identification on a real-world plate structure. For the sake of focused presentation of the current research, the detailed procedure for establishing the mathematical model of piezoelectric admittance-based damage identification, which can be found in literature (Shuai et al, 2017; Cao et al, 2023b), is outlined as shared data with link indicated in Appendix. Figure 6 illustrates the experimental setup, where a piezoelectric transducer is bonded to the top surface of a plate using epoxy resin. The parameters for the experimental setup are listed in Table 4. The plate is suspended by four thin strings at its corners to emulate free-free boundary conditions. A Dynamic Signal Analyzer (Agilent 35670A) measures the voltage across a 165 Ohm resistor to facilitate admittance measurement. A 5 Volts (Vpk) harmonic voltage with a good signal-to-noise ratio is applied to the transducer, and 25 cycles are used for each excitation frequency to reduce noise effects.



(a) (b)

Figure 6. Experimental setup (a) testbed; (b) schematical illustration of data acquisition.

Table 4. 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)	72.21
	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)	81.4
	Young’s modulus E_{33} (10^{3} MPa)	54
	Density (kg/m^3)	7800
	Piezo constant g_{31} (10^{-3} m^2/C)	-6.97
	Piezo constant g_{33} (10^{-3} m^2/C)	24.2
	Permittivity constant ε_{33} (10^{-8} F/m)	1.348

Concurrently, a finite element model is established using an in-house MATLAB code, verified with ANSYS. The plate and transducer are discretized using 20-node solid elements, resulting in the system comprising 13,776 elements and totaling 290,577 degrees of freedom. This is a very large scale problem, exemplifying the challenges in damage identification. The finite element model is then updated under the intact state based on experimental measurements, with result shown in Figure 7. To facilitate the identification process, the sensitivity approach by relating the admittance change to the damage index vector is adopted (Zhang et al, 2024a). Based on Section 2, we now have the damage identification model as:

$$\text{Find: } \boldsymbol{\alpha} \in \boldsymbol{E}^n,\ \alpha_l \le \alpha_i \le \alpha_u,\ i = 1, \ldots, n \tag{16a}$$

$$\text{Minimize: } f_1 = \|\Delta\mathbf{y} - \mathbf{T}\boldsymbol{\alpha}\|_2 \text{ and } f_2 = \|\boldsymbol{\alpha}\|_0 \tag{16b}$$

where $\Delta\mathbf{y}$ is admittance change from experimental measurements. $\mathbf{T}$ is sensitivity matrix calculated from the updated finite element model. $\boldsymbol{\alpha}$ is damage index vector. $\|\cdot\|_2$ and $\|\cdot\|_0$ denote, respectively, the l_2 norm and the l_0 norm. In this research we use the l_0 norm to achieve true sparsity since it directly penalizes the number of non-zero components.

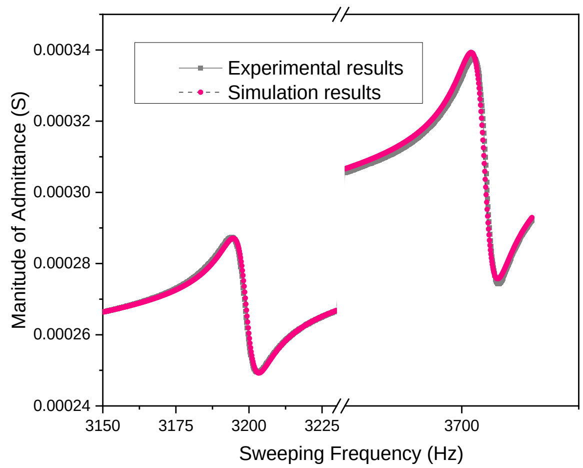


Figure 7. Admittance curves from experiment and from finite element analysis after model updating.

5.2.2 Damage identification practice employing LAMOPSO

Without loss of generality, we acquire admittance measurements for damage identification within two frequency ranges: 3,150 Hz to 3,230 Hz (covering the 28th natural frequency at 3,194.95 Hz) and 3,650 Hz to 3,730 Hz (covering the 31st natural frequency at 3,705.8 Hz). Figure 7 displays the experimentally acquired admittance curves of the healthy plate structure as a baseline. In order to identify damage, the plate structure being monitored is divided into 120 (12 lengthwise, denoted as L, and 10 widthwise, denoted as W) segments. Damage identification is facilitated through identifying the stiffness reduction at specific damaged segment(s). Therefore, the damage index vector $\boldsymbol{\alpha}$ has 120 unknowns. To ensure the repeatability and to avoid the alteration of boundary conditions, the structural damage is emulated by applying small mass blocks at various locations of the plate. Three experimental cases are designed as illustrated in Figure 8, 11.0% stiffness reduction in the 75$^{\text{th}}$ segment (Case 1); 9.4% stiffness reduction in the 116$^{\text{th}}$ segment (Case 2); 0.734% stiffness reduction in the 49$^{\text{th}}$ segment and 0.899% stiffness reduction in the 65$^{\text{th}}$ segment (Case 3).

To illustrate the inverse identification performance through multi-objective optimization, the proposed LAMOPSO is compared with MMOPSO and CMOQLMT algorithms, which exhibit competitiveness with the proposed algorithm on benchmark instances. All algorithms are executed with a population size of 50 and a maximum iteration number of 100,000 due to the increased number of variables. The results for the three cases are shown in Figures 9 and 10 with 3D bar plots, where each row shows the true damage scenario and the multiple solutions obtained from a specific algorithm. In each solution subplot, the location of a bar indicates the identified damage location, and the magnitude represents the calculated stiffness reduction.

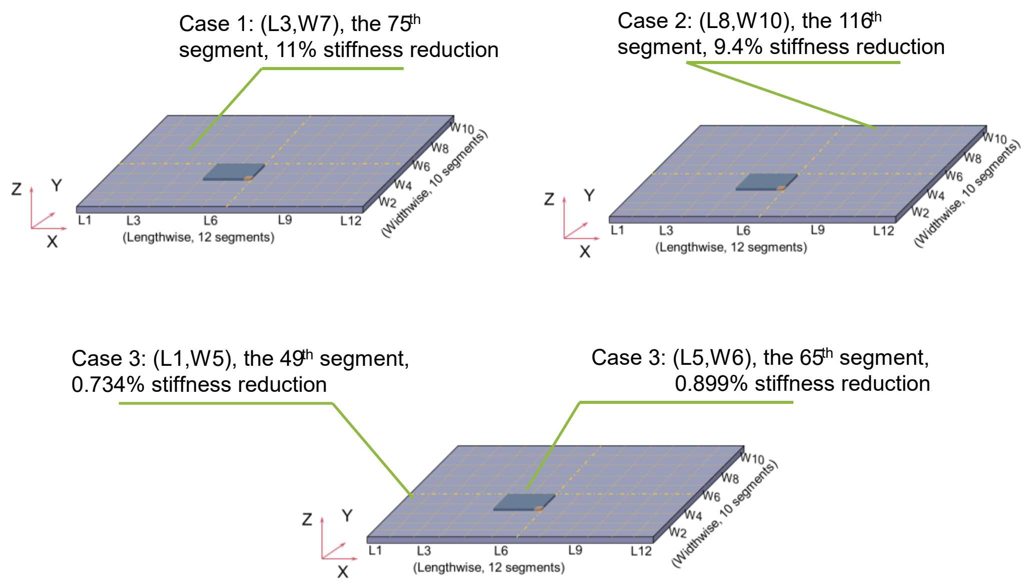


Figure 8. Illustration of segmentation and structural damage cases.

The results obtained for Case 1 are shown in Figure 9. The LAMOPSO, the MMOPSO, and the CMOQLMT find 2, 3, and 3 solutions, respectively. Through the multi-objective optimization formulation for inverse identification, we are able to obtain multiple solutions for an under-determined structural health monitoring problem, and the key is to find a small solution set. Apparently, all three algorithms are capable of achieving this in Case 1. Moreover, in each solution, only a small number of damage locations are identified. The maximum number of damage locations identified is 3 by the MMOPSO and CMOQLMT. Compared with the total number of segments 120, we can conclude that all three algorithms can produce sparse solutions. It is worth emphasizing that the sparsity of damage index vector reflects the nature of structural health monitoring where the number of damage locations is usually small especially during the early stage of damage occurrence.

The performance difference of these algorithms in Case 1 resides in solution accuracy and diversity. The LAMOPSO algorithm produces two solutions shown in the first row in Figure 9. The first solution effectively captures the true damage location, i.e., (L3, W7) or the 75th segment, with a damage severity of

10.27%, which is very close to the actual damage severity of 11.0% stiffness reduction. In the second solution, in addition to identifying the true damage location (the 75th segment) with severity 10.22%, the LAMOPSO identifies another damage location, the 10th segment with 0.24% stiffness reduction. As the severity in the 10th segment is very small compared with the true damage severity, we may deem it negligible. Essentially, these two solutions both point to the true damage scenario. The variant MMOPSO algorithm, in comparison, produces three solutions: 1) 12% stiffness reduction in the 95th segment; 2) 7.63% and 10.22% stiffness reductions in the 87th and the 95th segments, and 3) 7.58%, 10.15% and 0.12% stiffness reduction in the 87th, 95th, and 119th segments, respectively. None of these solutions points to the true damage scenario. Interestingly, the CMOQLMT algorithm produces three solutions with the solution 1 capturing the true damage scenario while the other two deviating from ground truth. The other two solutions obtained by the CMOQLMT algorithm identifies damage occurrence in the 10th segment and the 90th segment, and in the 10th, 63rd, as well as 90th segments, none of which points to the correct damage location. Overall, the proposed LAMOPSO algorithm outperforms the other two algorithms in terms of accuracy. It is worth noting that while the other two algorithms perform well in certain benchmark test cases as presented in Section 4.3, their performance in this structural damage identification problem with higher dimensionality becomes questionable.

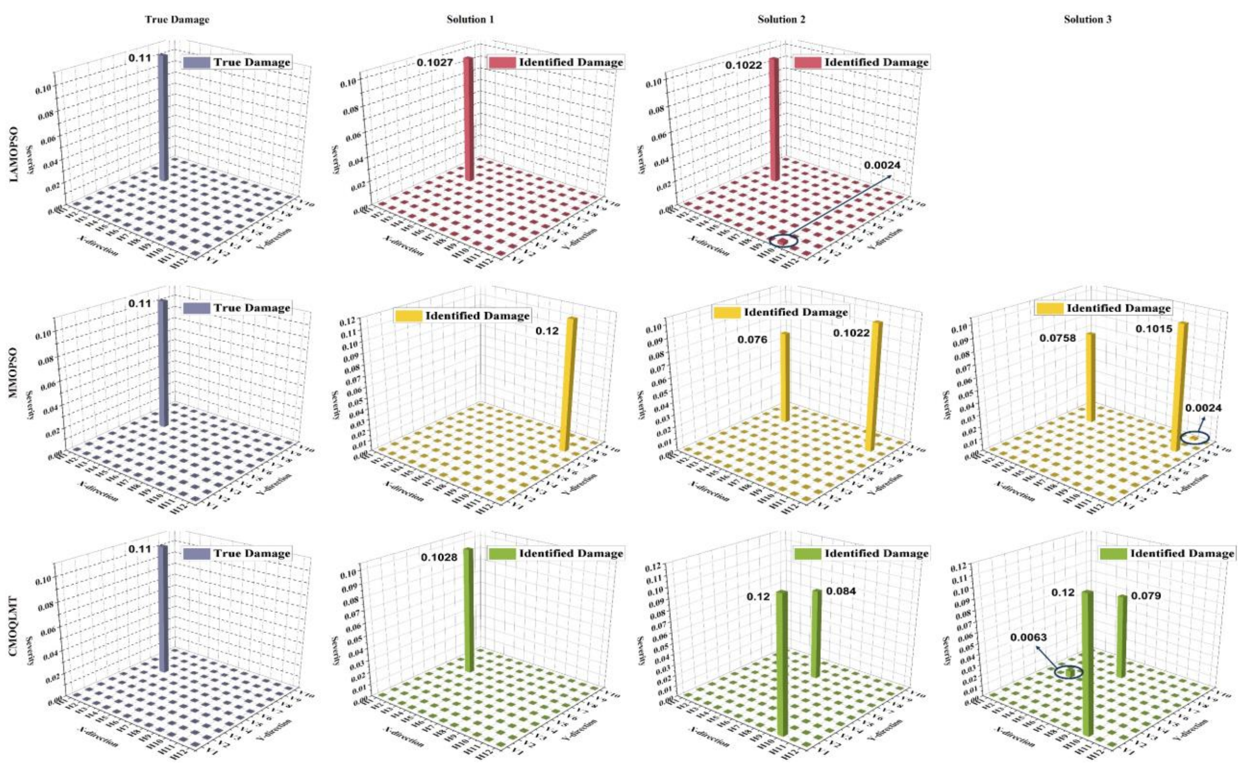


Figure 9. Inverse identification results of damage location/severity: Case 1.

In Case 2, the true damage location and severity are different from Case 1, and the solutions obtained from different algorithms are shown in Figure 10. All three algorithms produce 2 solutions. The solution set sizes are all small, which is good from structural health monitoring perspective. Each solution identified features sparsity, which is also good. Once again, the LAMOPSO algorithm produces two solutions shown in the first row in Figure 10. The first solution fits well the true damage location, i.e., (L8, W10) or the 116th segment, with a damage severity of 9.92%, which is close to the actual damage severity of 9.4% stiffness reduction. Similar to that in Case 1, in the second solution the LAMOPSO identifies two damage locations, with the first matching exactly the damage location (i.e., the 116th segment) with severity 8.96%, and the second in the 10th segment (L10, W1) with severity of 0.81%. In this solution, the second damage location has small severity, and this can be considered negligible when compared with the severity in the first location. Therefore, LAMOPSO can find two solutions that both point to the true damage scenario. The MMOPSO algorithm generates two solutions by none of them pinpointing the true damage location. Surprisingly, the CMOQLMT algorithm yields two solutions, and both capture the true damage scenario. The first solution points to the 116th segment with identified severity of 9.91% and the second solution identifies the 10th segment and the 116th segment with identified severity of 2.14% and 7.87% respectively. Overall, the proposed LAMOPSO and CMOQLMT algorithms achieve good performance on case 2. It is worth noting that both algorithms are improved with reinforcement learning, which means that the reinforcement learning shows promising potential in the optimizer enhancement.

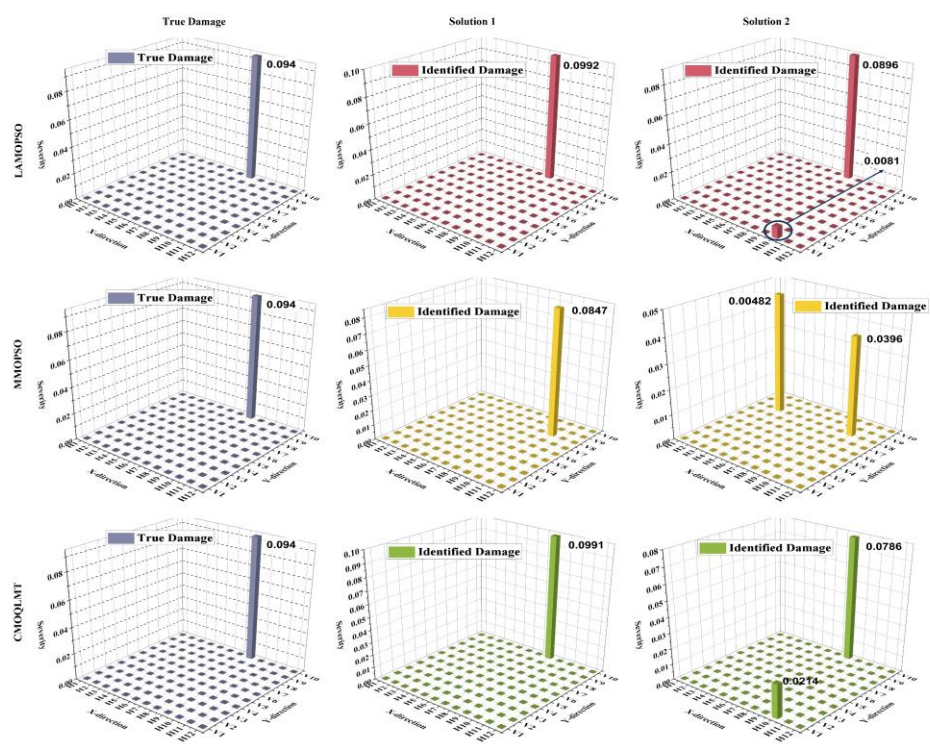


Figure 10. Inverse identification results of damage location/severity: Case 2.

Case 3 presents a more challenging problem with damage occurring in multiple locations with smaller severities. A well-performing algorithm would need to possess excellent global search capability. The results obtained from three algorithms are plotted in Figure 11. The maximum of solutions is 4 obtained by MMOPSO and CMOQLMT algorithms, while LAMOPSO yields 3 solutions. Although the true damage scenario has multiple damage locations in this case, it is not surprising for the algorithms to identify only one damage location as a possible solution because of the non-dominated sorting criterion involved in multi-objective optimization. The first solution by LAMOPSO does not match the true damage scenario. The second solution generated by the LAMOPSO algorithm captures the true damage scenario, pointing to the 49^{th} segment with 0.73% stiffness reduction and the 65^{th} segment with 0.907% stiffness reduction. The locations are correct, and the differences of damage severity with respect to true values, 0.734% and 0.899%, are extremely small. Furthermore, the third solution generated by the LAMOPSO is quite similar to the second solution, with additional damage location identified in the 105^{th} segment with a severity of 9.8e-6 that is indeed negligible. As such, both the second and the third solutions essentially match the true damage scenario in this very challenging case setup. In comparison, as shown in Figure 11, neither the MMOPSO nor the CMOQLMT can produce solutions that match the true damage scenario. It is apparent that both algorithms are trapped in local optima, pointing to locations that are different from the true locations. Overall, the proposed LAMOPSO algorithm beats the other two benchmark optimizers when dealing with the challenging damage identification problem.

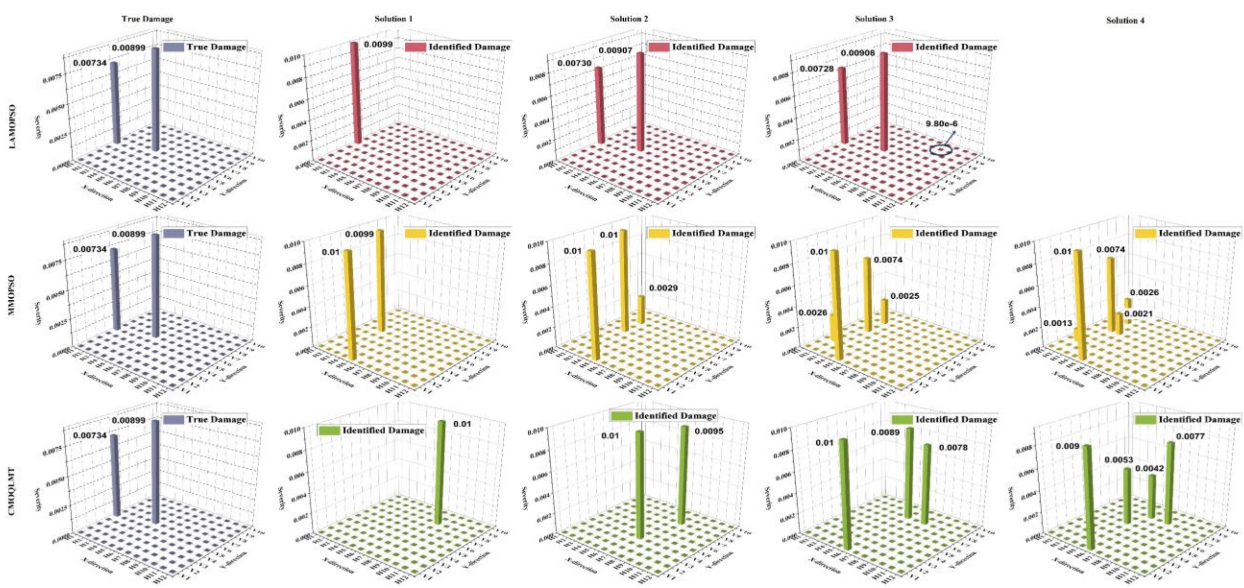


Figure 11. Inverse identification results of damage location/severity: Case 3.

5.3 Role of LA in enhancing MOPSO performance

The preceding subsection presents comprehensive case studies on the implementation of the proposed LAMOPSO algorithm to the challenging problem of structural damage identification using modal information or piezoelectric admittance measurement data. In these cases, the LAMOPSO performs better than the other two representative algorithms and leads to satisfying performance from the structural health monitoring perspective. Here we analyze the role of the learning automata (LA) in the proposed algorithm enhancement. In particular, as outlined in Section 3.3, 5 actions are implemented to guide the searching process. Their contributions are explicitly assessed in this subsection. In the proposed LAMOPSO algorithm, each particle is considered as a learning automaton. Each automaton has five selectable actions. Each action has a probability to indicate the likelihood that the action will be selected by the automaton. After the algorithm reaches convergence, we can visualize the final cumulative probability corresponding to each action for each learning automaton. To highlight the probability assessment in a repeatable manner with minimized influence of possible measurement noise and other uncertainties, here we use a numerical simulation case for illustration. We inject a damage with stiffness reduction of 3.4% to the 34th segment into the numerical model and conduct inverse identification employing simulation data. The population is set as 30 currently.

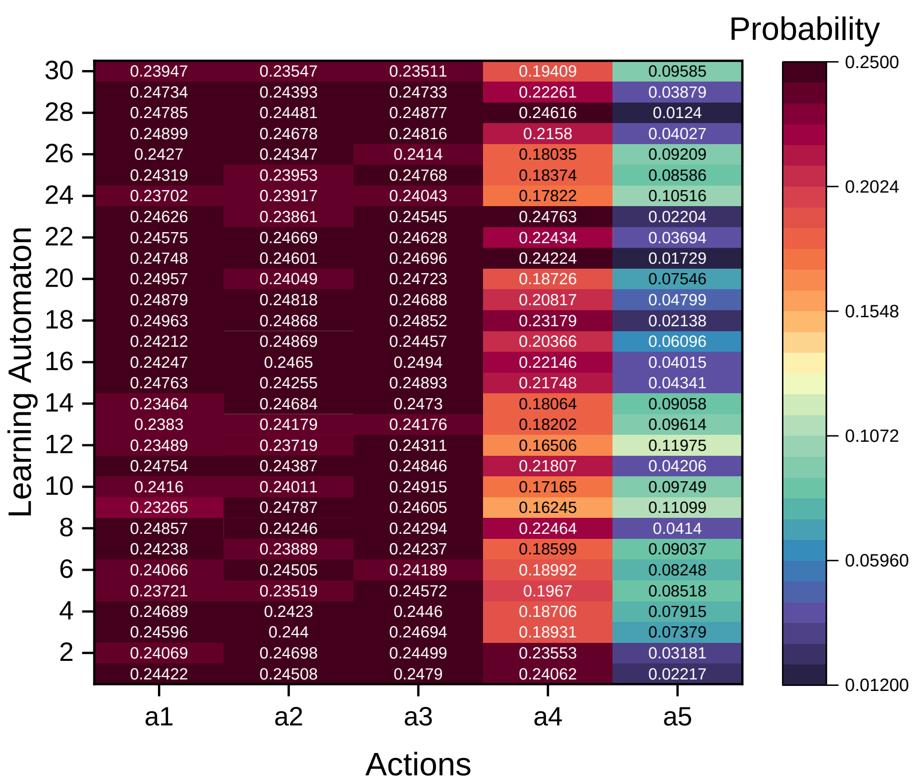


Figure 12. Probability distributions with 5 actions.

Shown in Figure 12 is the cumulative probability distribution plot. The horizontal axis corresponds to those 5 actions mentioned in Section 3.3, and the vertical axis corresponds to the 30 automatons (particles). In this figure, darker color indicates higher the probability of the corresponding action. We can observe that the final optimal actions are concentrated in the first three actions because they all possess a higher probability. Although the last two actions have cumulative probabilities, ultimately, they are not the optimal actions chosen by the learning automaton. As such, one question arises, i.e., why not remove the last two actions if they are not selected. To answer this question, we take a further look at the case where the last two actions are indeed removed. For that case, the cumulative probability distribution is shown in Figure 13. We can observe that the probability is nearly evenly distributed among 3 actions. However, the automatons still prefer the third action, as indicated by darker color. Therefore, we can conclude that the third action plays a very important role in updating the positions of particles and better solutions are obtained after executing this action. As formulated in Section 3, the third action is the mutation-based strategy for the personal best. This mutation strategy leads to diversity in the searching process.

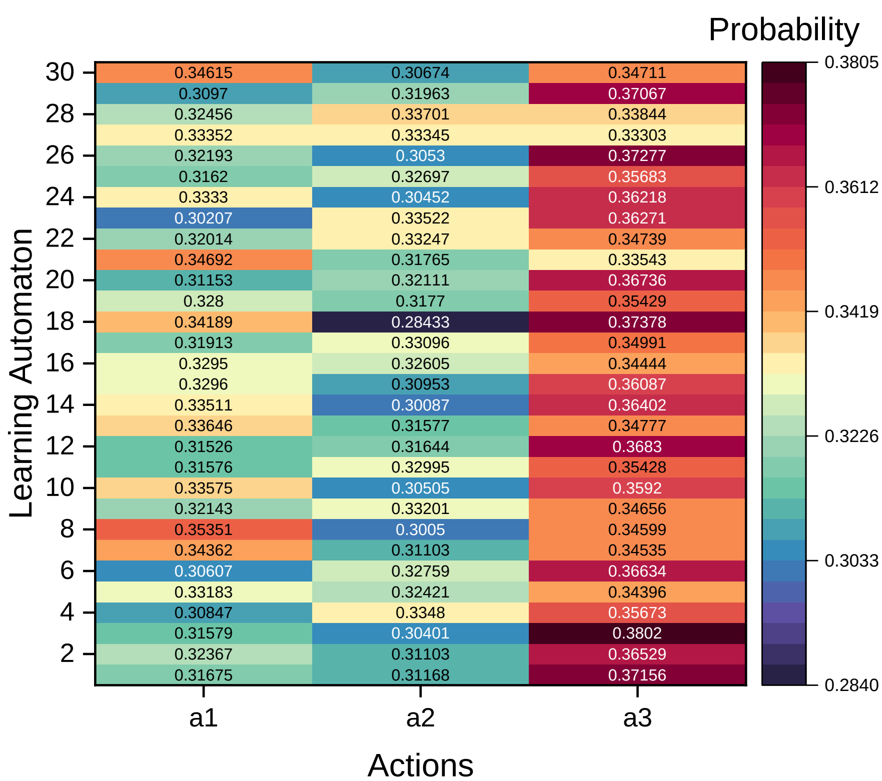


Figure 13. Probability distributions with 3 actions.

We can further examine how the last two actions are actually involved. We inspect the identification accuracy, i.e., the accuracy of stiffness reduction identified. Here we compare the true damage, the mean stiffness reduction corresponding to taking 5 actions and 3 actions respectively with 30 independent executions. To facilitate such comparison of damage involving location and severity, we define an *ad hoc* identification performance index. If the identification result points to a) the true damage location, and b) the stiffness reduction within 15% deviation from the true damage severity of 3.4%, then the result is considered *acceptable*. The identification performance index is defined as the ratio of the number of runs yielding acceptable result and the total number of runs (30). After 30 independent executions, the identification performance index is 93.33% with 5 actions, and 73.33% with 3 actions. Clearly, the last two actions play a very important role in ensuring the damage identification accuracy.

6 Conclusion and future work

This study presents a novel approach to structural damage identification through the development and implementation of a reinforcement learning-based multi-swarm optimizer. This optimizer is designed to address the complexities inherent in physical model-based damage identification, including the challenges posed by multimodality and high-dimensional variables. By integrating the particle swarm optimizer with metaheuristic strategies and leveraging the learning automata technique, we have successfully enabled an adaptive selection of search strategies that respond dynamically to the environmental context. This approach sidesteps the conventional need for ad hoc reward strategy selection, instead relying on the self-adaptation capabilities of learning automata to guide the particles' exploration of the search space effectively. The efficacy of proposed algorithm is underscored through rigorous validation against benchmark test cases and its application to real-world scenarios involving structural damage identification using modal information and piezoelectric admittance experimental signals. The results demonstrate the algorithm's ability to accurately identify a small, diverse solution set that closely corresponds to the true damage scenarios.

This research not only marks a significant advancement in the field of structural health monitoring but also opens new avenues for the application of swarm intelligence and reinforcement learning in complex optimization problems. It offers promising prospects for future research and practical applications in structural engineering, including dynamic optimization, path planning, traffic prediction, neural network structure optimization, structural design, and beyond. Besides, more diverse local search strategies can be developed and the deep Q-network, or related neural network model can be used for actions section.

Appendix

The multi-objective test functions and the piezoelectric admittance based damage identification method are posted online and can be accessed through this link: https://github.com/yantzan/PaperSubmissionSources/tree/c8c9ff1600dc0a794ad936eb25d7480138cb8b1d/ASC%20Resources

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 by a Space Technology Research Institutes grant (number 80NSSC19K1076) from NASA Space Technology Research Grants Program.

References

Avci O., Abdeljaber O., Kiranyaz S., Hussein M., Gabbouj M. and Inman D.J., A review of vibration-based damage detection in civil structures: From traditional methods to Machine Learning and Deep Learning applications. Mechanical systems and signal processing, 147 (2021) 107077.

Cao, P., Shuai, Q. and Tang, J., A multi-objective DIRECT algorithm toward structural damage identification with limited dynamic response information. Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems, 1(2) (2018) 021004-021004.

Cao P., Zhang Y., Zhou K., Tang J., A reinforcement learning hyper-heuristic in multi-objective optimization with application to structural damage identification, Structural and Multidisciplinary Optimization, 66(1) (2023a) 16.

Cao, P., Zhang, S., Wang, Z. and Zhou, K., Damage identification using piezoelectric electromechanical impedance: a brief review from a numerical framework perspective. Structures, 50 (2023b) 1906-1921.

Coello CAC, 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., 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, L., Wang, X., Yu, T. and Xia, Y., Vibration-based FRP debonding detection using a Q-learning evolutionary algorithm. Engineering Structures, 275 (2023) 115254.

Du Dinh-Cong, Trung Nguyen-Thoi, An effective damage identification procedure using model updating technique and multi-objective optimization algorithm for structures made of functionally graded materials. Engineering with Computers, 39 (2023) 1229–1247.

Ezzat, A.A., Tang, J. and Ding, Y., A model-based calibration approach for structural fault diagnosis using piezoelectric impedance measurements and a finite element model. Structural Health Monitoring, 19(6) (2020) 1839-1855.

Eberhart R and Kennedy J, Particle swarm optimization, In Proceedings of the IEEE international conference on neural networks, (4) (1995 November) 1942-1948.

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) (2019) 015003.

Friedrich T, Horoba C and Neumann F, Multiplicative approximations and the hypervolume indicator, In Proceedings of the 11th Annual conference on Genetic and evolutionary computation, (2009 July) 571-578.

Greś, S., Döhler, M. and Mevel, L., Statistical model-based optimization for damage extent quantification. Mechanical Systems and Signal Processing, 160 (2021) 107894.

Hashemi A B and Meybodi M R, A note on the learning automata based algorithms for adaptive parameter selection in PSO, Applied Soft Computing, 11(1) (2011) 689-705.

Ishibuchi H, Imada R, Setoguchi Y and Nojima Y, Reference point specification in inverted generational distance for triangular linear Pareto front. IEEE Transactions on Evolutionary Computation, 22(6) (2018) 961-975.

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

Li Y. F., Minh, H.L., Khatir, S., Sang-To, T., Cuong-Le, T., MaoSen, C. and Wahab, M.A., Structure damage identification in dams using sparse polynomial chaos expansion combined with hybrid K-means clustering optimizer and genetic algorithm. Engineering Structures, 283 (2023) 115891.

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

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.

Lin, Q., Li, J., Du, Z., Chen, J. and Ming, Z., A novel multi-objective particle swarm optimization with multiple search strategies. European Journal of Operational Research, 247(3) (2015) 732-744.

Lin, Q., Liu, S., Zhu, Q., Tang, C., Song, R., Chen, J., Coello, C.A.C., Wong, K.C. and Zhang, J., 2016. Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems. IEEE Transactions on Evolutionary Computation, 22(1) (2018) 32-46.

Ming, F., Gong, W. and Gao, L., Adaptive Auxiliary Task Selection for Multitasking-Assisted Constrained Multi-Objective Optimization [Feature]. IEEE Computational Intelligence Magazine, 18(2), (2023) 18-30.

Noel, M.M., 2012. A new gradient based particle swarm optimization algorithm for accurate computation of global minimum. Applied Soft Computing, 12(1), pp.353-359.

Ratnaweera A, Halgamuge S K and Watson H C, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Transactions on Evolutionary Computation, 8(3) (2004) 240-255.

Shi Y and Eberhart R C, Empirical study of particle swarm optimization. In Proceedings of the 1999 congress on evolutionary computation (CEC), (3) (1999) 1945-1950.

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.

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.

Thanh Sang-To, Hoang Le-Minh, Seyedali Mirjalili, Magd Abdel Wahab, Thanh Cuong-Le, A new movement strategy of grey wolf optimizer for optimization problems and structural damage identification, Advances in Engineering Software, 173 (2022) 103276.

Tian, Y., Cheng, R., Zhang, X. and Jin, Y., PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. IEEE Computational Intelligence Magazine, 12(4), (2017) 73-87.

Tian Y, Cheng R, Zhang X, Li M and Jin Y, Diversity assessment of multi-objective evolutionary algorithms: Performance metric and benchmark problems [research frontier], IEEE Computational Intelligence Magazine, 14(3) (2019) 61-74.

Tian, Y., Li, X., Ma, H., Zhang, X., Tan, K.C. and Jin, Y., Deep reinforcement learning based adaptive operator selection for evolutionary multi-objective optimization. IEEE Transactions on Emerging Topics in Computational Intelligence, 7(4) (2023) 1051-1064.

Yue, C., Qu, B. and Liang, J., A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems. IEEE Transactions on Evolutionary Computation, 22(5) (2018) 805-817.

Zitzler E, Deb K and Thiele L, Comparison of multiobjective evolutionary algorithms: Empirical results, Evolutionary Computation, 8(2) (2000) 173-195.

Zhang Q, Zhou A, Zhao S, Suganthan PN, Liu W, Tiwari S, Multiobjective optimization test instances for the CEC 2009 special session and competition. University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report, (2008) 264.

Zhang Z, Wang D and Gao J, Learning automata-based multiagent reinforcement learning for optimization of cooperative tasks, IEEE Transactions on Neural Networks and Learning Systems, 32(10) (2020) 4639-4652.

Zhou K. and Tang J., Structural model updating using adaptive multi-response Gaussian process meta-modeling. Mechanical Systems and Signal Processing, 147 (2021) 107121.

Zhang Z. and Sun C., Structural damage identification via physics-guided machine learning: a methodology integrating pattern recognition with finite element model updating. Structural Health Monitoring, 20(4) (2021) 1675-1688.

Zhang, Y., Zhou, K. and Tang, J., Piezoelectric impedance-based high-accuracy damage identification using sparsity conscious multi-objective optimization inverse analysis. Mechanical Systems and Signal Processing, 209 (2024a) 111093.

Zhang, G., Kang, J., Wan, C., Xie, L. and Xue, S., Output-only structural damage identification based on Q-learning hybrid evolutionary algorithm and response reconstruction technique. Measurement, 224 (2024b) 113951.