

RESEARCH ARTICLE

Investigation of parameters of the modified fractional derivative constitutive model for viscoelastic dampers using various meta-heuristic optimization techniques

Suhail Alabsi^{1*} , Ayşe T. Daloğlu¹ , Sameer Alabsi¹ 

¹ Karadeniz Technical University, Faculty of Engineering, Department of Civil Engineering, Trabzon, Türkiye

Article History

Received 5 November 2024

Accepted 26 December 2024

Keywords

Viscoelastic dampers

Modified Fractional Derivative
Constitutive Model

Non-dominated Sorting Genetic
Algorithm II

Teaching–Learning–Based
Optimization

Harmonic Search Algorithm

Particle Swarm Optimization

Abstract

This study aims to improve the parameters of the modified fractional derivative constitutive model (MFDCM) for viscoelastic dampers (VEDs), as these parameters are extremely important in reducing and resisting structural responses to dynamic loads, such as seismic and wind loads. The MFDCM, with its nonlinear and frequency-dependent characteristics, is a complex model, which directly affects the storage modulus (G_s) and loss factor (η) of VEDs, leading to great difficulty in accurately predicting the damper behavior under different conditions. The problem studied is inherently multi-objective, involving trade-offs between errors in the storage modulus (G_s) and loss factor (η). First, a multi-objective approach is employed to identify a set of potential solutions and generate a Pareto front, which provides insights into the trade-offs between competing objectives. Precise weights are then determined from the Pareto front to transform the multi-objective problem into a single-objective problem, allowing further refinement using single-objective optimization techniques. Four advanced meta-heuristic optimization techniques—Non-dominated Sorting Genetic Algorithm II (NSGA-II), Teaching–Learning–Based Optimization (TLBO), Particle Swarm Optimization (PSO), and Harmony Search Algorithm (HSA)—are employed to systematically reduce the error rates in the storage modulus (G_s) and loss factor (η) predictions compared to experimental data. The results of this study demonstrate that, by incorporating multiple optimization techniques, the prediction accuracy of the MFDCM can be significantly enhanced. This improved modeling ability thus enables better design of VEDs, improving their performance and reliability in practical engineering applications. Comparative analysis of different algorithms provides insights into their effectiveness and efficiency, offering valuable guidance for choosing appropriate optimization strategies in engineering optimization problems.

* Corresponding author (suhail00alabsi@gmail.com)

eISSN 2630-5763 © 2023 Authors. Publishing services by [golden light publishing®](https://goldenlightpublishing.com).

This is an open-access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Optimization is becoming more important in many areas of science and is now a key part of fields like engineering, economics, and management [1]. It helps solve complex problems in logistics, decision-making, and system design, and it is widely used in operations research, computer science, and engineering [2]. Over time, researchers have developed different methods to find solutions, and recent advances in technology and the growing complexity of real-world problems have made optimization even more necessary [3]. These techniques are now commonly used in engineering to solve a variety of challenges in areas like materials science and computer systems [4,5]. However, choosing the best optimization technique can still be difficult, especially for problems that are very complex or have many variables [6,7].

The modified fractional derivative constitutive model (MFDCM) is a useful tool for understanding how viscoelastic dampers (VEDs) behave under different conditions [8]. VEDs are important for reducing the impact of dynamic loads, such as those caused by earthquakes and wind loads. Despite its usefulness, the MFDCM is challenging to optimize because of its complexity and reliance on experimental data. This makes it hard to improve the model's accuracy in predicting how VEDs perform in real situations. To address this issue, advanced optimization techniques are needed to handle the trade-offs between different objectives. Many studies have shown how optimization techniques can improve engineering solutions [9–13]. For instance, Altun et al. [14] compared five algorithms to show how different methods perform under various conditions. Jain et al. [15] tested Multi-Objective Particle Swarm Optimization (MOPSO), Non-Dominated Sorting Genetic Algorithm II (NSGA-II), and other techniques to balance costs and emissions in power systems. Wen et al. [16] introduced an improved version of NSGA-II to solve scheduling problems, proving its effectiveness. Kashani et al. [17] highlighted how NSGA-II and MOPSO can optimize the design of retaining walls, showing NSGA-II's ability to manage competing goals like cost and safety. Other studies have used similar methods to solve complex problems, further proving their value [18–25].

This research focuses on improving the parameters of the MFDCM using four optimization methods: Non-dominated Sorting Genetic Algorithm II (NSGA-II), Teaching–Learning-Based Optimization (TLBO), Particle Swarm Optimization (PSO), and Harmony Search Algorithm (HSA). First, a multi-objective approach is used to explore different solutions, and then weights from the Pareto front are applied to convert the problem into a single-objective one for further optimization. This approach ensures that the parameters are adjusted to achieve better model accuracy. [26] By applying these techniques to experimental data, this study provides a clear comparison of their performance and offers guidance for improving models of VEDs in practical engineering applications.[27].

2. Modified fractional derivative constitutive model (MFDCM)

Fig. 1. shows the structure of the modified fractional derivative constitutive model (MFDCM). This model consists of two main components: an elastic spring and a fractional Kelvin element. The elastic spring represents the solid-like, elastic behavior of the material, while the fractional Kelvin element accounts for the time-dependent, viscous properties of viscoelastic materials (VEMs). Together, these elements enable the MFDCM to capture both elastic and viscoelastic responses under varying loading conditions.

MFDCM offers a significant advantage over conventional integer derivative models. Its ability to describe the behavior of VEMs across a wide frequency range with fewer parameters makes it particularly efficient and accurate [28,29]. This capability reduces the complexity of parameter fitting, enhancing its applicability in modeling real-world materials.

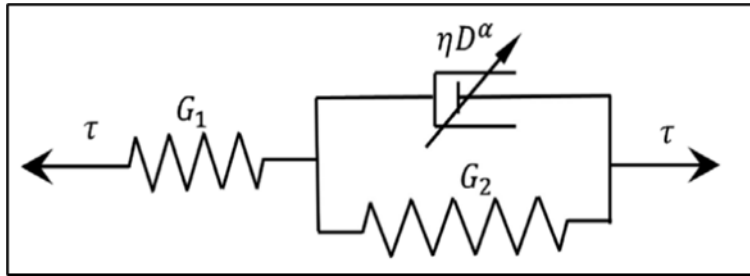


Fig. 1. Modified fractional derivative constitutive model (MFDCM) [11]

MFDCM offers a significant advantage over conventional integer derivative models. Its ability to describe the behavior of VEMs across a wide frequency range with fewer parameters makes it particularly efficient and accurate [28,29]. This capability reduces the complexity of parameter fitting, enhancing its applicability in modeling real-world materials.

$$\tau(t) + p_1 D^\alpha [\tau(t)] = q_0 \gamma(t) + q_1 D^\alpha [\gamma(t)] \quad (1)$$

where p_1 , q_0 , and q_1 the coefficients related to the VEMs, that can be determined by experimental data. $\tau(t)$ and $\gamma(t)$ are the shear stress and strain at physical time t , respectively. D^α is the fractional derivative operator with a power of α , where $0 < \alpha < 1$, at value $\alpha = 1$, the model considers as classic three-parameter model.

MFDCM provides a comprehensive framework to characterize the behavior of viscoelastic dampers (VEDs) under varying dynamic conditions. The governing equations of the model account for the nonlinear and frequency-dependent properties of the system, essential for accurately predicting the performance of VEDs. The storage modulus (G_s) and loss factor (η). are described as:

$$G_s(\omega) = \frac{q_0 + p_1 q_1 \omega^{2\alpha} + (q_0 p_1 + q_1) \omega^\alpha \cos \frac{\pi\alpha}{2}}{1 + p_1^2 \omega_\alpha^2 + 2p_1 \omega^\alpha \cos \frac{\pi\alpha}{2}} \quad (2)$$

$$\eta(\omega) = \frac{(q_1 - q_0 p_1) \omega^\alpha \sin \frac{\pi\alpha}{2}}{q_0 + p_1 q_1 \omega^{2\alpha} + (q_0 p_1 + q_1) \omega^\alpha \cos \frac{\pi\alpha}{2}} \quad (3)$$

where ω represents the circular frequency of excitation.

To adapt these equations for practical applications, the coefficients q_0 , q_1 , and p_1 are reformulated in terms of displacement amplitude as follows:

$$q_0^{ref} = \frac{G_1 G_2}{G_1 + G_2} \quad (4)$$

$$q_1^{ref} = \frac{G_1 \eta}{G_1 + G_2} \quad (5)$$

$$q_0 = q_0^{ref} \lambda_1 \quad (6)$$

$$q_1 = q_1^{ref} \lambda_2 \quad (7)$$

$$\lambda_1 = 1 + e^{-C_1 \gamma_{ref}} \quad (8)$$

$$\lambda_2 = 1 + e^{-C_2 \gamma_{ref}} \quad (9)$$

The parameters q_0^{ref} and q_1^{ref} represent the original values derived from the standard fractional derivative model. The terms G_1 and G_2 denote the shear modulus of the respective components in the system, while η indicates the viscosity of the fractional dashpot. The coefficients C_1 and C_2 govern the softening rates for q_0 and q_1 under varying shear strains.

The reference shear strain γ_{ref} is defined as:

$$\gamma_{ref} = \frac{u_0}{t_v} \quad (10)$$

where, u_0 is the maximum displacement of the intermediate steel plate and t_v is the thickness of the VEM layer.

These equations form the foundation for modeling and optimizing the parameters of the MFDCM, ensuring that it captures the nonlinearities and dynamic responses of VEDs with precision. By incorporating these refined formulations, the model aligns with the primary objective of improving the accuracy and reliability of VED simulations under real-world conditions.

3. Problem definition and optimization framework

The modified fractional derivative constitutive model (MFDCM) parameters play a crucial role in accurately representing the behavior of viscoelastic dampers (VEDs). This includes defining the objectives, error functions, and constraints, as well as outlining the approach used to address these through optimization techniques.

3.1. Problem statement

The error functions are critical in assessing the accuracy of the modified fractional derivative constitutive model (MFDCM) by measuring the differences between the model-predicted values and experimental data for two important viscoelastic properties: the storage modulus G_s and the loss factor η . Minimizing these discrepancies ensures that the model reliably reflects real-world material behavior under dynamic loading conditions. The use of these error functions varies depending on whether a single-objective or multi-objective optimization framework is applied.

3.2. Error functions for multi-objective optimization

In multi-objective optimization, the goal is to simultaneously minimize two independent error functions: the storage modulus error (**Error** $_{G_s}$) and the loss factor error (**Error** $_{\eta}$). These objectives are inherently conflicting, as reducing the error in one parameter may lead to an increase in the error of the other.

3.2.1. Storage modulus error (**Error** $_{G_s}$)

Quantifies the absolute difference between the predicted storage modulus G_s^{model} and the experimental storage modulus G_s^{target} across all frequencies:

$$Error_{G_s} = \sum_{\omega} |G_s^{model}(\omega) - G_s^{target}(\omega)| \quad (11)$$

$$G_s^{model}(\omega) = \lambda_1 \cdot \frac{q_0 + p_1 q_1 \omega^{2\alpha} + (q_0 p_1 + q_1) \omega^{\alpha} \cos \frac{\pi\alpha}{2}}{1 + p_1^2 \omega_{\alpha}^2 + 2p_1 \omega^{\alpha} \cos \frac{\pi\alpha}{2}} \quad (12)$$

where $Error_{G_s}$ The total error in the storage modulus across all frequencies, $G_s^{model}(\omega)$ The predicted value of the storage modulus from the model at a specific frequency (ω), $G_s^{target}(\omega)$ the experimental target value of the storage modulus at the same frequency (ω), ω The angular frequency of the excitation.

3.2.2. Loss factor error ($Error_\eta$)

Quantifies the absolute difference between the predicted loss factor (η^{model}) and the experimental loss factor (η^{target}).

$$Error_\eta = \sum_{\omega} |\eta^{model}(\omega) - \eta^{target}(\omega)| \quad (13)$$

$$\eta^{model}(\omega) = \lambda_2 \cdot \frac{(q_1 - q_0 p_1) \omega^\alpha \sin \frac{\pi \alpha}{2}}{q_0 + p_1 q_1 \omega^{2\alpha} + (q_0 p_1 + q_1) \omega^\alpha \cos \frac{\pi \alpha}{2}} \quad (14)$$

where $Error_\eta$ The total error in the loss factor across all frequencies, $\eta^{model}(\omega)$ The predicted value of the loss factor from the model at a specific frequency (ω), $\eta^{target}(\omega)$ the experimental target value of the loss factor at the same frequency (ω), ω The angular frequency of the excitation.

For each frequency, the model predictions are computed using the equations outlined above. The absolute differences between the predicted and experimental values are summed across all frequencies to compute the total errors for storage modulus (G_s) and loss factor (η).

3.3. Error function for single-objective optimization

In single-objective optimization, the storage modulus error ($Error_{G_s}$) and the loss factor error ($Error_\eta$) are combined into a single fitness function. This transformation simplifies the optimization process and allows the use of efficient single-objective optimization algorithms.

This transformation simplifies the optimization process and allows for the application of efficient single-objective optimization. The fitness function is constructed as a weighted summation of the two error functions, enabling the balancing of conflicting objectives by adjusting their relative importance. The fitness function is expressed mathematically as:

$$F = W_{G_s} \cdot Error_{G_s} + W_\eta \cdot Error_\eta \quad (15)$$

where W_{G_s} is the weight assigned to the storage modulus error and W_η is the weight assigned to the loss factor error.

These weights represent the relative importance of each objective in the context of the application. By varying weights, the optimization process can prioritize one objective over the other, such as emphasizing stiffness (G_s) or energy dissipation (η), or achieving a balanced trade-off. This single fitness function acts as the objective for the optimization process. Combining the errors into a single scalar value enhances computational efficiency, enabling Teaching–Learning–Based Optimization (TLBO), Particle Swarm Optimization (PSO), and Harmony Search Algorithm (HSA) to converge rapidly to an optimal solution. This approach also offers flexibility, as modifying the weights allows the optimization process to adapt to specific performance goals or application priorities.

3.4. Justification for multi-objective and single-objective optimization

Using both multi-objective and single-objective optimization approaches in this study ensures a thorough and effective parameter optimization process for the modified fractional derivative constitutive model

(MFDCM). Each method brings distinct advantages, and their combined use is crucial for obtaining accurate and dependable results.

3.4.1. Multi-objective optimization

Multi-objective optimization is especially effective for addressing problems with conflicting objectives, such as simultaneously minimizing the errors in the storage modulus (G_s) and loss factor (η). These two parameters often embody competing performance criteria in modeling viscoelastic dampers (VEDs). Employing a multi-objective optimization, such as Non-dominated Sorting Genetic Algorithm II (NSGA-II), facilitates the generation of a Pareto front, offering a comprehensive visualization of the trade-offs between G_s and η [30].

The Pareto front provides engineers and researchers with a spectrum of optimal solutions, enabling flexibility in choosing the most suitable solution based on specific application requirements. For example, some applications may prioritize energy dissipation (η) over stiffness (G_s), while others may require a balance between the two. Additionally, multi-objective optimization offers valuable insights into parameter interactions, revealing how variations in one parameter impact the other [31]. This comprehensive perspective is essential for understanding the behavior of the modified fractional derivative constitutive model (MFDCM) and ensuring that the optimization process effectively addresses all relevant trade-offs.

3.4.2. Single-objective optimization

While multi-objective optimization is crucial for examining trade-offs, single-objective optimization plays a key role in fine-tuning and meeting specific performance targets. By consolidating the errors in storage modulus (G_s) and loss factor (η) into a single fitness function, the optimization process becomes streamlined and focused, enabling efficient convergence toward the desired performance outcomes. Single-objective optimization simplifies the problem and enables efficient parameter adjustments [32]. The weighting mechanism weight assigned to the storage modulus (W_{G_s}) and the weight assigned to the loss factor error (W_η) provides the flexibility to prioritize objectives according to specific application needs. For instance, assigning a higher weight to W_{G_s} places greater emphasis on reducing the error in the storage modulus, while increasing W_η focuses on minimizing the loss factor error. This adaptability ensures that the optimization aligns with engineering requirements, whether the goal is maximizing stiffness, enhancing damping, or achieving a balanced trade-off between the two [33].

In addition to flexibility, single-objective optimization offers computational efficiency. Algorithms like Teaching–Learning-Based Optimization (TLBO), Particle Swarm Optimization (PSO), and Harmony Search Algorithm (HSA) can quickly converge to an optimal solution due to the simplicity of optimizing a single fitness function. This rapid convergence makes single-objective optimization particularly effective for refining solutions identified through multi-objective approaches [34,35].

The integration of multi-objective and single-objective optimization ensures a comprehensive framework for parameter optimization in this study. Multi-objective optimization explores the solution space and identifies Pareto-optimal solutions, highlighting trade-offs between G_s and η , which inform the weighting scheme for single-objective optimization. Single-objective optimization then refines these solutions for precise parameter adjustments. This hybrid approach combines the broad exploration of multi-objective methods with the targeted refinement of single-objective techniques, achieving accurate and reliable parameter estimation for engineering applications involving viscoelastic dampers (VEDs).

4. Parameter boundaries and weight selection for single-objective optimization

In this study, the parameter boundaries and weight values for single-objective optimization were meticulously defined to ensure consistency, physical relevance, and applicability to viscoelastic dampers

(VEDs). These configurations were uniformly applied across all optimization techniques Teaching–Learning-Based Optimization (TLBO), Particle Swarm Optimization (PSO), and Harmony Search Algorithm (HSA)—to maintain a standardized and comparable framework.

4.1. Weight selection

The weights $W_{G_s} = 1$ and $W_\eta = 5.5$ were determined as optimal through extensive testing and insights gained from multi-objective optimization using Non-dominated Sorting Genetic Algorithm II (NSGA-II). The Pareto front generated by NSGA-II revealed the trade-offs between storage modulus (G_s) and loss factor (η), facilitating an informed selection of weights to balance these conflicting objectives.

The chosen weights reflect the study's practical needs, prioritizing $W_\eta = 5.5$ to ensure the accuracy of η , which is highly sensitive to variations and crucial for capturing the damping performance of viscoelastic dampers (VEDs). In contrast, $W_{G_s} = 1$ assigns a lower weight to the G_s , maintaining balance without compromising its accuracy.

This deliberate selection process highlights the effort to align the weights with the study's practical requirements rather than relying on arbitrary values. The selected weights provide a balanced optimization framework that captures the engineering importance of both objectives effectively.

4.2. Parameter boundaries

The parameter boundaries were selected to ensure physical relevance to viscoelastic dampers (VEDs), and adherence to prior research, thereby guaranteeing realistic and meaningful optimization results. The fractional derivative parameter α is bound between 0 and 1, as suggested by foundational studies on fractional derivatives. This range ensures that the damping behavior modeled by α remains physically meaningful and aligns with theoretical expectations.

The material parameters (C_1, C_2, G_1, G_2) are constrained within 0.01 to 1000, reflecting experimentally observed ranges for viscoelastic dampers. These bounds encompass the practical ranges of material properties influencing both the storage modulus (G_s) and loss factor (η).

The friction parameter is restricted to values between 0.01 and 1, representing the typical range for viscoelastic materials (VEMs) used in damping systems. This ensures that the frictional component remains realistic and accurately represents the materials under investigation.

5. Non-dominated sorting genetic algorithm 2 (NSGA-II)

The Non-dominated Sorting Genetic Algorithm II (NSGA-II) is a sophisticated algorithm designed for multi-objective optimization, offering effective solutions to complex optimization problems. Originally introduced by Deb et al. [36], it operates on the principle of selecting dominant solutions among a set of potential candidates. The flowchart of the NSGA-II is shown in Fig.2 [37]. A brief overview of this algorithm's approach is provided as follows [38].

- a) Initialization of population P_0 of size N using a uniform distribution.
- b) Generate new offspring population Q_t by utilizing binary tournament selection which is based on crowding comparison operator, crossover, and mutation operation on the parent population (P_t). Here, t denotes the number of generations. The entire population R_t is the combination of the offspring population Q_t and its parent population P_t .
- c) Non-dominated fronts of different objective functions are obtained by performing a fast non-dominated sorting approach on the entire population (R_t).
- d) Generate a new parent population (P_{t+1}) from the obtained fronts.
- e) This process is continued until the maximum number of iterations is reached.

The Non-dominated Sorting Genetic Algorithm II (NSGA-II) was implemented in MATLAB R2024b, leveraging its robust optimization toolbox to handle the complexities of multi-objective optimization. MATLAB's platform offered the flexibility to define objective functions $Errorr_{G_s}$ and $Errorr_{\eta}$, set parameter boundaries and establish stopping criteria. Furthermore, it facilitated the efficient processing of large datasets and iterative computations, ensuring accurate and computationally efficient optimization results.

The population size was set to 50, a value determined after extensive experimentation with varying sizes. Initial tests with smaller populations of 20 and 30 individuals showed limited diversity, which restricted the algorithm's ability to thoroughly explore the Pareto front. By increasing the population size to 50, a balance was achieved, enabling more comprehensive coverage of the solution space while maintaining computational efficiency.

The maximum number of generations was set to 5,000, a value determined through iterative testing. Trials with smaller limits, such as 2,000 and 3,000 generations, resulted in incomplete convergence, failing to capture the full trade-offs between $Errorr_{G_s}$ and $Errorr_{\eta}$. Extending the limit to 5,000 ensured stabilizations of the Pareto front, providing a comprehensive understanding of the trade-offs between the conflicting objectives.

The function tolerance was set to 1×10^{-9} , ensuring that the optimization process considered only significant improvements in the objective values, thereby maintaining precision, and avoiding convergence to suboptimal solutions. Additionally, the maximum stall generations parameter was configured to 5,000, effectively preventing premature termination caused by stagnation, and allowing the algorithm sufficient opportunity to explore and refine the Pareto front.

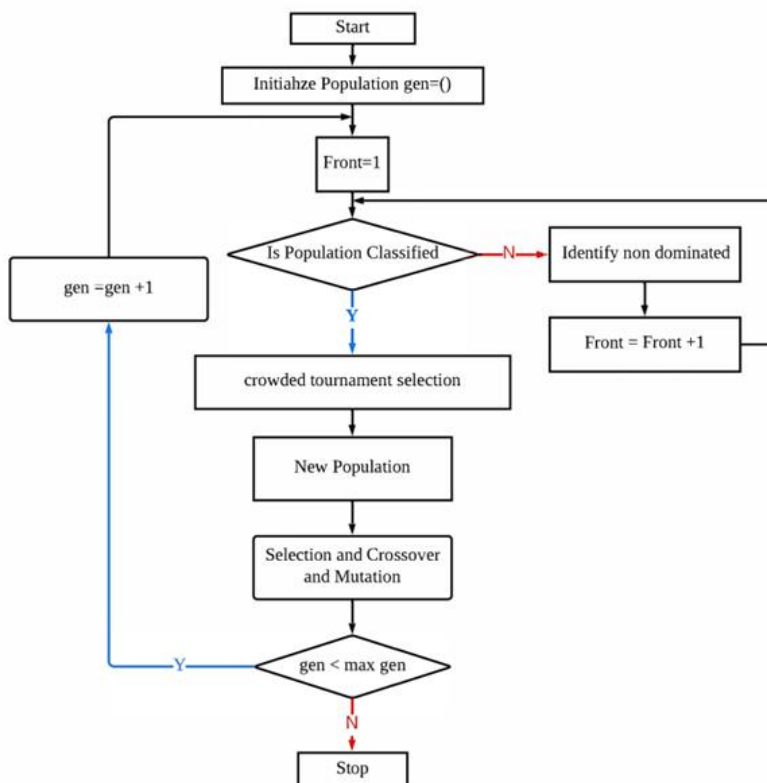


Fig. 2. Flowchart of NSGA-II [37]

The crossover and mutation operations were strategically configured to preserve diversity within the population while simultaneously refining promising solutions. A simulated binary crossover (SBX) operator, with a distribution index of 20, was employed to ensure effective exploration of the search space. For mutation, a polynomial mutation technique was applied, using a distribution index of 20 and a mutation probability of $1/n_{vars}$, where n_{vars} represents the number of variables (six in this study).

Additionally, MATLAB's built-in visualization tools were leveraged to dynamically plot and analyze the Pareto front. This real-time monitoring capability enabled the identification of optimization trends and facilitated parameter adjustments as needed, ensuring a robust and effective optimization process.

NSGA-II generated a Pareto front that effectively visualized the trade-offs between $Error_{G_s}$ and $Error_{\eta}$. This Pareto front, depicted, offered critical insights into the relative importance of minimizing errors in the storage modulus (G_s) and loss factor (η).

The solutions along the Pareto front were carefully analyzed to derive the weights W_{G_s} and W_{η} for the single-objective optimization phase. This multi-objective analysis provided a foundation for the subsequent optimization steps, enabling a balanced approach that addressed both objectives comprehensively.

6. Harmony search algorithm (HSA)

The Harmony Search Algorithm (HSA) was first introduced by Zong Woo Geem in 2001 [39]. Inspired by the improvisation process used by musicians, HSA mimics how musicians try to find the best harmony by playing different notes. This analogy is applied to optimization problems, where different solutions are generated and improved iteratively to find the best solution. The algorithm works through three key operations: memory consideration, pitch adjustment, and randomization. Over the years, HSA has been successfully applied in various fields, particularly in engineering optimization problems, such as material optimization.

The flowchart of the Harmony Search Algorithm is shown in Fig. 3 [40], which provides a visual representation of the key steps in the algorithm, including initialization, improvisation, memory update, and termination. For a comprehensive explanation of this methodology, readers are encouraged to refer to the foundational references [40, 41].

The Harmony Search Algorithm (HSA) was implemented using MATLAB R2024b, with meticulously chosen parameters to achieve an optimization process that is both effective and efficient. These parameters were iteratively refined through extensive testing of various configurations to strike an optimal balance between computational cost and accuracy. The use of MATLAB R2024b facilitated the simulation of complex scenarios and enabled the fine-tuning of parameters within a robust computational environment. The parameter boundaries were carefully defined to remain within physically meaningful and positive ranges, ensuring consistency with the practical characteristics of viscoelastic dampers (VEDs) and guaranteeing the reliability of the optimization results.

The Harmony Memory Size (HMS) was set to 50 after iterative testing with smaller values, such as 5 and 10, which demonstrated limited exploration capabilities and often resulted in suboptimal solutions. By gradually increasing the HMS, the algorithm gained the ability to explore the solution space more thoroughly. At $HMS = 50$, the optimization process showed significant improvements in solution quality without substantially increasing computational costs. Further increases beyond this point yielded diminishing returns, solidifying $HMS = 50$ as the optimal choice for balancing efficiency and performance.

The Harmony Memory Consideration Rate (HMCR) was configured at 0.9 to ensure a high likelihood of selecting values from the existing Harmony Memory (HM). This setting allowed the algorithm to predominantly utilize promising solutions stored in HM while preserving a degree of randomness necessary for effective exploration. The high HMCR value proved crucial in leveraging the stored information

efficiently, striking a balance between exploiting good solutions and maintaining sufficient diversity to prevent premature convergence.

The Pitch Adjustment Rate (PAR) was set to 0.3 to achieve a balance between exploration and exploitation. A lower PAR led to inadequate refinement of the harmonies, limiting the algorithm's ability to fine-tune solutions. Conversely, higher PAR values caused excessive disruptions, particularly in stable regions of the solution space, hindering convergence. The chosen value of 0.3 provided an optimal trade-off, enabling precise local adjustments around promising solutions while maintaining overall stability in the optimization process.

The Bandwidth (bw) was set to 0.05 to allow for fine-grained adjustments during the pitch adjustment phase. This narrow adjustment range was specifically chosen to address the sensitivity of the modified fractional derivative constitutive model's (MFDCM) parameters, as even small variations can significantly influence the model's performance. By using this carefully selected bandwidth, the algorithm was able to refine solutions with precision, ensuring that the optimization process captured the intricate dynamics of the model while avoiding overshooting optimal parameter values.

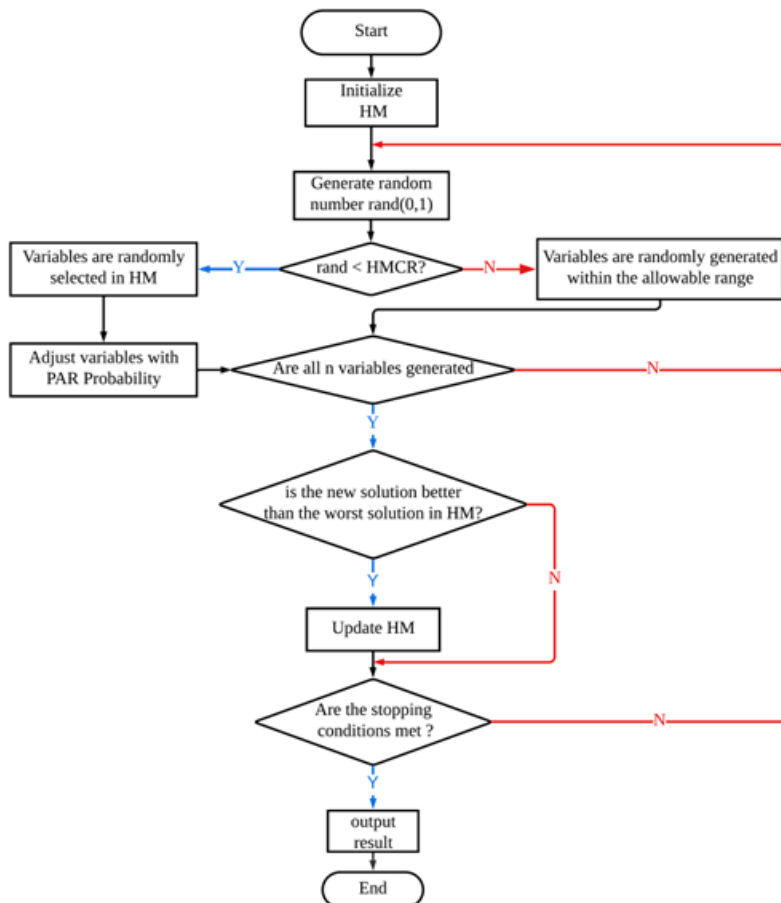


Fig. 3. Flowchart of HSA [40]

The Maximum Iterations were determined to be 20,000 after an extensive analysis of the relationship between the number of iterations and error reduction. Early tests with 5,000 and 10,000 iterations revealed that the algorithm had not fully converged, leaving room for further improvement. Increasing the iterations to 20,000 allowed the optimization process to consistently achieve convergence and yield optimal results. Beyond this threshold, additional iterations offered negligible improvements, validating the efficiency and sufficiency of this setting for capturing the model's optimal parameters.

7. Teaching-learning-based optimization (TLBO)

The Teaching-Learning-Based Optimization (TLBO), introduced by Rao et al. [42], is a population-based optimization technique inspired by the teaching-learning in classroom. This algorithm divides the optimization process into two main phases: the Teacher Phase and the Learner Phase As illustrated in Fig. 4 [42].

In the Teacher Phase, the teacher represents the best solution found so far, to increase the mean level of knowledge (or fitness) of the learners (population). The goal is to shift the class mean, M , toward the teacher's level, denoted by $X_{teacher}$. The mean difference is calculated as:

$$DM = r \times (M_{new} - TF \times M_{old}) \quad (16)$$

where M_{old} is the current mean, M_{new} is the new target mean defined by the teacher's influence, TF is the teaching factor, randomly set to 1 or 2 to control the extent of influence, r is a random number between 0 and 1. so this update for each learner X_i in this phase is given by

$$X_{new,i} = X_{old,i} + D_F \quad (17)$$

In the Learner Phase, learners improve through peer-to-peer interactions. Each learner randomly selects another learner and updates their position based on their fitness values. If learner X_i has a lower fitness than the selected peer, X_j then X_i is updated as:

$$X_{new,i} = X_{old,i} + r \times (X_j - X_i) \quad (18)$$

If X_i has a better fitness, then the update equation is

$$X_{new,i} = X_{old,i} + r \times (X_i - X_j) \quad (19)$$

This peer exchange enables diversity in the population, helping avoid local minima by allowing learners to adopt parts of their peers' solutions.

Teaching-Learning-Based Optimization (TLBO) for the modified fractional derivative constitutive model (MFDCM) was efficiently implemented using MATLAB R2024b. This setup allowed for optimal computational efficiency and precise parameter tuning through iterative adjustments. MATLAB's capabilities facilitated the handling of complex characteristics inherent to the MFDCM, ensuring an effective balance between computational speed and accuracy of the optimization process.

The population size was configured to be 60 learners, a value determined through iterative testing with smaller and larger populations. Initial experiments with 20 and 40 learners demonstrated inadequate diversity, which often resulted in premature convergence to suboptimal solutions. Expanding the population size to 60 facilitated a more comprehensive exploration of the search space while preserving computational efficiency, striking a balance between solution quality and processing time.

The maximum number of generations was determined to be 10,000 after iterative testing. Early trials with 2,000 and 5,000 generations showed incomplete convergence, with the algorithm failing to achieve optimal parameter settings. Extending the number of generations to 10,000 allowed sufficient iterations for the error

rate to stabilize, ensuring the reliability and accuracy of the results. This choice provided a balance between computational cost and achieving robust optimization outcomes.

During the early stages (generations 1–1,000), the algorithm utilized a higher level of randomness ($\sigma = 0.7$) to promote extensive global exploration and avoid premature convergence. In the mid-stage (generations 1,001–3,000), randomness was reduced to $\sigma = 0.3$, enabling the algorithm to concentrate on refining solutions around promising areas identified earlier. In the final stage (generations 3,001–10,000), randomness was further minimized to $\sigma = 0.05$, allowing for precise fine-tuning of the solutions to achieve optimal parameter settings.

The teaching factor was dynamically set to either 1 or 2, based on the algorithm's internal logic. This adaptive setting allowed the teacher to effectively guide the learners, enhancing their convergence toward improved solutions by balancing exploration and exploitation throughout the optimization process.

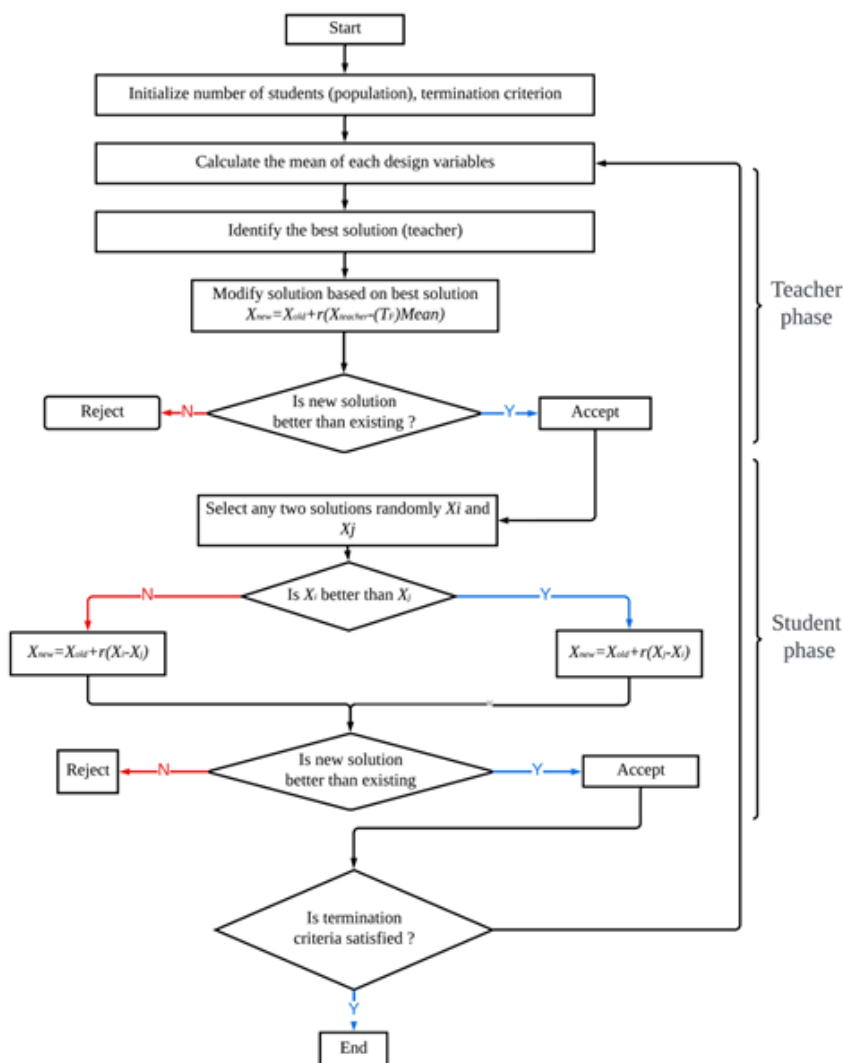


Fig. 4. Flowchart of TLBO [42]

To maintain feasibility, all new solutions generated during the teacher and learner phases were restricted within the predefined parameter boundaries. These boundaries, consistently applied across all optimization algorithms, were designed to reflect the physical relevance of the viscoelastic dampers (VEDs). parameters, ensuring that the optimization results remained valid and practically applicable.

8. Particle swarm optimization (PSO)

The Particle Swarm Optimization (PSO) was introduced by James Kennedy and Russell Eberhart in 1995 [43], as explained by Genovesi et al. [20], PSO is an evolutionary approach to the collective behavior seen in bird flocks. In PSO, each particle stands for a possible solution and moves through a multi-dimensional space affected by three main factors: its momentum, feedback from its personal best position, and guidance from the swarm's best position found so far. The flowchart of the PSO is illustrated in Fig. 5 [44]. The equation for this update is given as [44,45]:

$$v_{i+1} = w \cdot v_i + c_1 \cdot r_1 \cdot (p_{best} - x_i) + c_2 \cdot r_2 \cdot (g_{best} - x_i) \quad (20)$$

Where w is the inertia weight, determining the degree to which previous velocity is retained, c_1 is the cognitive coefficient, which weighs the particle's personal best position p_{best} , c_2 is the social coefficient, which weighs the global best position g_{best} discovered by the swarm, r_1 and r_2 are random factors between 0 and 1, introducing randomness to help avoid local minima.

The new position of the particle is then updated using:

$$x_{i+1} = x_i + v_{i+1} \quad (21)$$

By balancing exploration (global search) and exploitation (local refinement), this multi-step iteration of velocity and position updates enables particles to converge toward the optimal solution. The PSO preserves simplicity and efficiency while searching the search space, which makes it particularly useful for problems involving continuous variables.

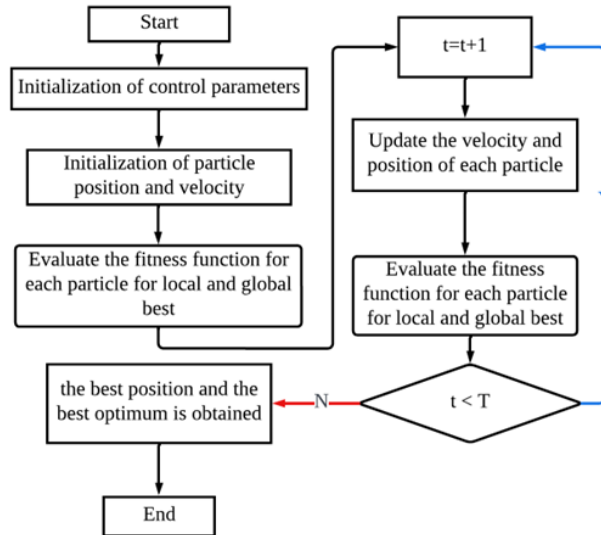


Fig. 5. Flowchart of PSO [44]

To optimize the computational efficiency of the Particle Swarm Optimization (PSO) implementation, we utilized MATLAB's parallel computing capabilities from the outset. By setting '*UseParallel*', *true*, the optimization process was executed across multiple cores, specifically leveraging the eight-core configuration of our computing system. This parallel execution allowed for simultaneous function evaluations, significantly speeding up the convergence process by enabling a more extensive search within a reduced timeframe.

PSO was implemented with carefully selected parameters to ensure a balance between effectiveness and computational efficiency. These parameters were iteratively adjusted and refined through experimental trials, aiming to achieve reliable convergence while minimizing computational overhead. This approach ensured that the optimization process remained both accurate and practically applicable to engineering challenges.

The swarm size was configured to 50 particles, determined through iterative testing. Initial experiments with smaller sizes, such as 10 and 20, revealed limited search space exploration, often resulting in premature convergence. By increasing the swarm size to 50, the algorithm enhanced its ability to explore the solution space comprehensively, achieving a balance between computational efficiency and solution quality. Further increases in swarm size provided minimal gains in performance while substantially raising computation time, affirming 50 as the optimal choice.

The maximum number of iterations was configured to be 200 based on an in-depth analysis of the relationship between iteration count and error reduction. Initial trials with 100 iterations showed incomplete convergence, highlighting the need for additional iterations to refine the solutions. Increasing the iteration limit to 200 consistently led to optimal results, with the error stabilizing effectively. Further increases in the iteration count provided negligible improvements, confirming that 200 iterations were sufficient to achieve convergence and optimize the solution efficiently.

The inertia weight (ω) was dynamically varied throughout the optimization process to balance exploration and exploitation effectively. It was initialized at 0.9 to promote extensive global exploration during the early stages of optimization, enabling the algorithm to traverse a wide solution space. As the process advanced, ω was gradually reduced to 0.4, shifting the focus toward local exploitation to refine solutions around promising regions. This adaptive approach improved the algorithm's efficiency and significantly enhanced its ability to identify optimal solutions by leveraging both broad search capabilities and fine-tuning adjustments.

9. Case study

9.1. Experimental data overview

The experimental performance data used in this study were sourced from benchmark research conducted by T. Zhang et al. [8]. This data forms the cornerstone for the parameter optimization of the modified fractional derivative constitutive model (MFDCM). The experimental setup, depicted in Fig. 6, was designed to measure the force–displacement hysteresis behavior of viscoelastic dampers (VEDs) under cyclic loading. This setup effectively highlights the energy dissipation and stiffness characteristics of VEDs across a range of frequencies and amplitudes.

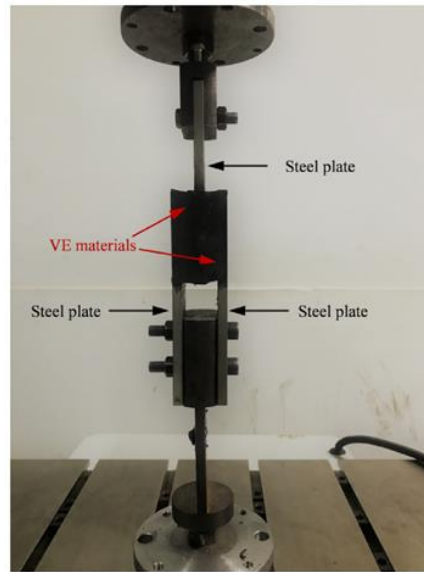


Fig. 6. VED test setup [8]

9.2. Force–displacement hysteresis and optimization relevance

The force–displacement hysteresis loops, depicted in Fig. 7, offer valuable insights into the dynamic behavior of viscoelastic dampers (VEDs). These loops illustrate the force-displacement relationship under varying frequencies and amplitudes, enabling the derivation of critical viscoelastic materials (VEMs) properties. These properties, extracted from the elliptical representation of the hysteresis loops (as shown in Fig. 8), formed the basis for defining the optimization objectives: the storage modulus (G_s) and loss factor (η). Rather than focusing on the experimental procedures, the emphasis is on the utilization of these hysteresis data for optimization.

The extracted parameters were utilized as benchmarks for minimizing errors during the optimization process. These calculated error functions informed the application of single-objective optimization methods Teaching–Learning-Based Optimization (TLBO), Particle Swarm Optimization (PSO), and Harmony Search Algorithm (HSA)—and the multi-objective optimization method Non-dominated Sorting Genetic Algorithm II (NSGA-II), to effectively tune the MFDCM's parameters. By integrating these experimental data, the optimization framework closely mirrors the physical behavior of VED, thereby enhancing the model's reliability and predictive accuracy.

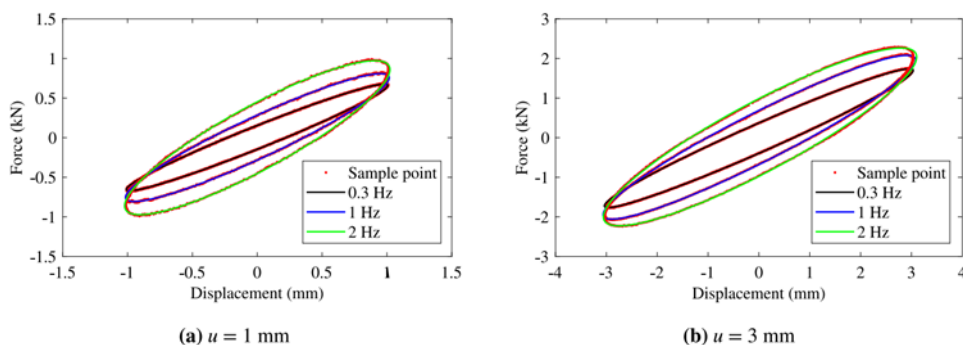


Fig. 7. Force–displacement hysteresis curves of viscoelastic damper (VED). at identical displacement amplitudes but varying frequencies [8]

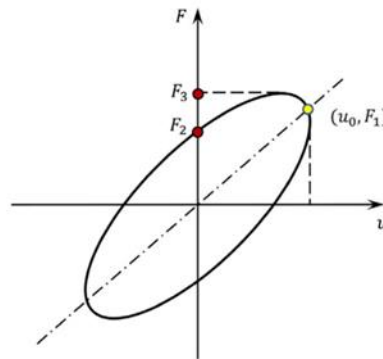


Fig. 8. Characteristic hysteresis loop of the experimental VED [8]

10. Results and discussion

To evaluate the effectiveness of the modified fractional derivative constitutive model (MFDCM) for the viscoelastic damper (VED), the experimental results were compared with the model's calculated outcomes, as shown in Table 1. The comparison highlights two key dynamic properties: storage modulus (G_s) and loss factor (η). Additionally, to illustrate the effectiveness of the meta-heuristic optimization used in this study, the results are compared with those from Zhang et al. [8] who employed the Multi-Objective Goal Attainment Optimization as displayed in Table 1.

The parameters of the MFDCM were identified through a combination of experimental data and model fitting, as shown in Table 2. These six parameters C_1 , C_2 , G_1 , G_2 , α , and η play a critical role in defining VED behavior under varying frequencies and shear strains. Here G_1 and G_2 represent stiffness contribution of the respective components in the MFDCM. Higher values of G_1 indicate a stiffer system, while G_2 contributes to the overall energy dissipation. η represents the viscosity parameter that controls the rate of energy dissipation during cyclic loading. Higher values of η indicate improved damping capabilities. α is the fractional derivative order, which adjusts the frequency dependency of the MFDCM. Values close to 1 imply a more linear response, while lower values suggest more complex VED behavior. C_1 , C_2 are coefficients that adjust the softening behavior of the VED under different shear strains. These parameters ensure that MFDCM captures the nonlinearity observed in experimental data, especially under high shear conditions.

Table 3. provides error percentages for each opt Optimization Techniques. The errors were determined by comparing the calculated values from the model with the corresponding experimental data. For all optimization techniques, the highest error for the G_s occurs at a frequency of 2 Hz and a shear strain of 0.3. Similarly, the worst-case scenario for the η is observed at a frequency of 0.3 Hz and a shear strain of 0.3. The consistency in the relationship between error values and the frequency-shear strain combination suggests that this behavior is linked to the MFDCM.

According to Zhang et al. [8], the maximum relative error between the experimental and model calculated values for the storage modulus G_s is 11.86%, while for the loss factor η , it reaches 15.35%. In contrast, the meta-heuristic optimization techniques proposed in this study significantly improved MFDCM by reducing the largest relative error for the storage modulus G_s to 11.41%, 10.66%, 11.16%, and 11.12%, for NSGA-II, TLBO, HSA, and PSO, respectively. Similarly, the largest relative error for the loss factor η was reduced to 14.89%, 14.77%, 12.68%, and 12.77% for NSGA-II, TLBO, HSA, and PSO, respectively. Relatively TLBO demonstrated the best relative error for the storage modulus, add to HSA demonstrated the best relative error for loss factor modulus.

Table 1. Storage modulus and Loss Factor comparison between experimental and the MFDCM results

Frequency (Hz)	Shear Strain	Storage modulus G_S (MPa)						Loss Factor η					
		Experiment	Goal attainment [8]	NSGA-II	TLBO	HSA	PSO	Experiment	Goal attainment [8]	NSGA-II	TLBO	HSA	PSO
0.3	0.05	1.136	1.099	1.1345	1.1097	1.1003	1.099	0.232	0.222	0.2237	0.2238	0.2278	0.228
1	0.05	1.298	1.291	1.3311	1.3016	1.2949	1.294	0.384	0.413	0.4127	0.4089	0.4117	0.412
2	0.05	1.473	1.509	1.5479	1.5089	1.5032	1.502	0.542	0.542	0.5457	0.5411	0.5418	0.542
0.3	0.1	1.089	1.058	1.0837	1.0650	1.0580	1.057	0.231	0.213	0.2134	0.2140	0.2185	0.218
1	0.1	1.235	1.235	1.2629	1.2412	1.2375	1.237	0.378	0.398	0.3963	0.3936	0.3973	0.397
2	0.1	1.399	1.437	1.4604	1.4313	1.4295	1.429	0.526	0.526	0.5270	0.5235	0.5255	0.525
0.3	0.15	1.050	1.021	1.0386	1.0248	1.0197	1.019	0.227	0.205	0.2052	0.2060	0.2107	0.211
1	0.15	1.189	1.185	1.2037	1.1880	1.1866	1.186	0.369	0.386	0.3832	0.3809	0.3852	0.385
2	0.15	1.325	1.373	1.3856	1.3642	1.3652	1.364	0.522	0.512	0.5118	0.5088	0.5116	0.511
0.3	0.2	1.005	0.987	0.9982	0.9886	0.9851	0.985	0.228	0.199	0.1988	0.1996	0.2045	0.204
1	0.2	1.142	1.141	1.1520	1.1411	1.1415	1.141	0.365	0.376	0.3729	0.3707	0.3753	0.375
2	0.2	1.242	1.317	1.3215	1.3058	1.3089	1.308	0.515	0.500	0.4998	0.4969	0.5002	0.500
0.3	0.25	0.973	0.956	0.9621	0.9558	0.9536	0.953	0.227	0.194	0.1940	0.1947	0.1995	0.199
1	0.25	1.101	1.101	1.1068	1.0996	1.1013	1.101	0.356	0.368	0.3651	0.3627	0.3674	0.367
2	0.25	1.176	1.267	1.2662	1.2549	1.2593	1.259	0.499	0.490	0.4907	0.4875	0.4909	0.490
0.3	0.3	0.943	0.927	0.9297	0.9261	0.9249	0.925	0.224	0.190	0.1906	0.1909	0.1956	0.195
1	0.3	1.062	1.066	1.0670	1.0627	1.0653	1.065	0.346	0.361	0.3596	0.3566	0.3612	0.361
2	0.3	1.094	1.223	1.2184	1.2102	1.2156	1.215	0.478	0.483	0.4842	0.4803	0.4836	0.483

Table 2. Parameters of the MFDCM

	G_1 (MPa)	G_2 (MPa)	η (MPa s^a)	α	C_1	C_2
Goal attainment [8]	5.4488	0.5449	0.1224	0.6836	1.3508	3.4651
NSGA-II	8.0143	0.5451	0.1247	0.6649	1.6433	4.2354
TLBO	9.6399	0.5205	0.1198	0.6520	1.4363	3.8764
HSA	9.998	0.5102	0.1207	0.6451	1.3582	3.6259
PSO	9.988	0.5096	0.1207	0.6450	1.3488	3.6368

Table 3. Error comparison for different optimization techniques

Frequency (HZ)	Shear Strain	Storage modulus G_s (error value %)					Loss Factor η (error value %)				
		Goal attainment [8]	NSGA-II	TLBO	HSA	PSO	Goal attainment [8]	NSGA-II	TLBO	HSA	PSO
0.3	0.05	3.24	0.14	2.32	3.15	3.23	4.43	3.54	3.51	1.78	1.75
1	0.05	0.56	2.54	0.27	0.24	0.32	7.39	7.36	6.37	7.09	7.11
2	0.05	2.44	5.05	2.41	2.02	1.95	0.02	0.69	0.15	0.01	0.01
0.3	0.1	2.80	0.45	2.17	2.81	2.88	7.84	7.47	7.21	5.27	5.27
1	0.1	0.01	2.27	0.51	0.21	0.14	5.49	4.99	4.26	5.24	5.22
2	0.1	2.68	4.37	2.30	2.17	2.10	0.01	0.28	0.38	0.01	0.03
0.3	0.15	2.83	1.14	2.44	2.93	2.98	9.81	9.70	9.33	7.24	7.27
1	0.15	0.31	1.23	0.08	0.20	0.25	4.68	3.93	3.31	4.48	4.44
2	0.15	3.58	4.54	2.93	3.00	2.94	1.92	1.88	2.45	1.92	1.96
0.3	0.2	1.86	0.72	1.68	2.03	2.07	12.91	12.84	12.48	10.36	10.41
1	0.2	0.13	0.83	0.12	0.09	0.13	3.04	2.25	1.65	2.92	2.85
2	0.2	5.98	6.36	5.09	5.34	5.29	2.93	2.95	3.51	2.88	2.95
0.3	0.25	1.82	1.17	1.82	2.05	2.07	14.51	14.33	14.05	11.93	12.00
1	0.25	0.00	0.50	0.16	0.00	0.04	3.18	2.51	1.82	3.14	3.05
2	0.25	7.71	7.62	6.66	7.04	6.99	1.71	1.64	2.29	1.61	1.69
0.3	0.3	1.69	1.45	1.83	1.95	1.97	15.3	14.89	14.77	12.68	12.77
1	0.3	0.31	0.42	0.02	0.27	0.24	4.42	4.01	3.15	4.48	4.37
2	0.3	11.87	11.41	10.66	11.16	11.12	1.08	1.37	0.56	1.26	1.16

10.1. Results for harmonic search algorithm (HSA)

The Harmonic Search Algorithm (HSA) showcased exceptional effectiveness in optimizing the parameters of the modified fractional derivative constitutive model (MFDCM). The algorithm minimized discrepancies in both storage modulus (G_s) and loss factor (η), demonstrating its capability to enhance the model's accuracy and dependability.

Through the application of weights W_{G_s} and W_η the optimization process achieved a substantial reduction in errors in Fig. 9. Starting at an initial error rate of 39.6% in the first iteration, the process saw a dramatic decline to 7.75% by the 627th iteration. The optimization stabilized by the 16,136th iteration, achieving a final error rate of 2.1% for the objective function.

The error reduction trend, depicted in Fig. 9, clearly illustrates the algorithm's efficiency in addressing the optimization challenge, with a steady decline in error rates signifying effective parameter tuning and convergence.

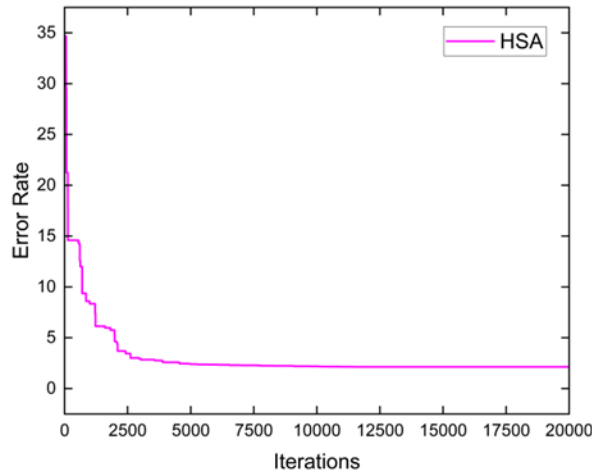


Fig. 9. Iterations vs. average error HSA

10.2. Results for particle swarm optimization (PSO)

Particle Swarm Optimization (PSO) demonstrated strong effectiveness in optimizing the parameters of the modified fractional derivative constitutive model (MFDCM). Its rapid exploration of the parameter space and efficient convergence highlight its capability to minimize errors in both storage modulus (G_s) and loss factor (η). The optimized parameter values for the MFDCM, obtained through PSO, are shown in Table 2. These results align with the physical behavior of viscoelastic dampers and have been successfully validated using experimental data.

A swarm size of 50 exhibited a strong performance. During the initial iterations, a significant reduction in the error rate was observed, as illustrated in Fig. 10. This trend indicates that PSO effectively explores the parameter space and quickly identifies regions with improved solutions. As iterations progress, minor fluctuations in the error rate occur, reflecting the particles' exploration of various areas in the solution space, which may temporarily lead to local optima. The error stabilizes at 1.92% after 191 iterations. This discussion highlights that PSO outperforms the other optimization methods studied, both in minimizing the total error rate and in reducing computational time.

The optimized parameters obtained through PSO for the MFDCM, as presented in Table 2, show strong alignment with the physical properties of viscoelastic dampers. These parameters have been thoroughly validated against experimental data, confirming their accuracy and relevance for modeling the behavior of Viscoelastic dampers (VEDs) under dynamic conditions.

10.3. Results for teaching–learning-based optimization (TLBO)

The Teaching–Learning-Based Optimization (TLBO) demonstrated significant effectiveness in optimizing the modified fractional derivative constitutive model (MFDCM). By leveraging its structured optimization approach, TLBO successfully minimized errors in both storage modulus (G_s) and loss factor (η), while maintaining computational efficiency.

Implemented with a population size of 60, TLBO ensured adequate diversity to explore the parameter space comprehensively. The error vs. generation plot, illustrated in Fig. 11, depicts the optimization trajectory. During the initial generations, a rapid reduction in error was observed, driven by the algorithm's capacity to exploit the random initial population and promptly identify superior solutions. This early convergence highlights TLBO's ability to refine solutions effectively during the initial stages of the optimization process.

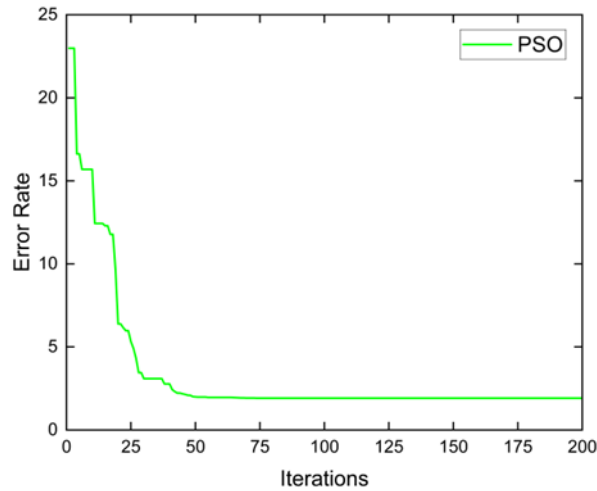


Fig. 10. Iterations vs. average error PSO

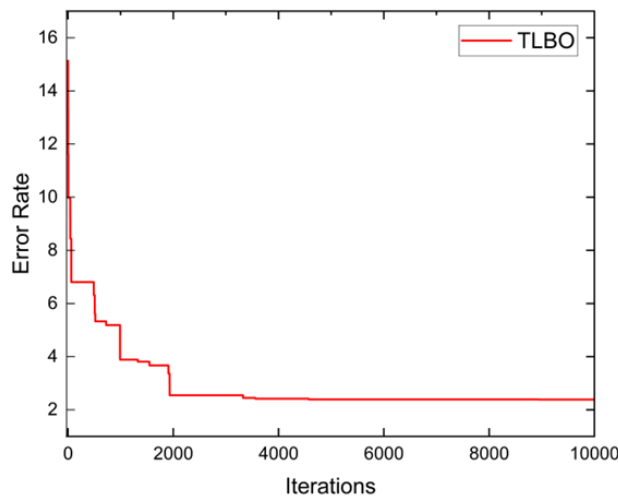


Fig. 11. Iterations vs. average error TLBO

As the generations advanced, the rate of improvement gradually diminished, indicating the algorithm's convergence towards optimal or near-optimal solutions. By the 8,597th generation, the error stabilized at 2.4%, signifying that further improvements were either negligible or had ceased entirely, marking the conclusion of the optimization process.

10.4. Results for non-dominated sorting genetic algorithm ii (NSGA-II)

The Non-dominated Sorting Genetic Algorithm II (NSGA-II) was utilized to optimize the parameters of the modified fractional derivative constitutive model (MFDCM). By addressing the conflicting objectives of minimizing $Error_{G_s}$ and $Error_{\eta}$, NSGA-II effectively generated a set of Pareto-optimal solutions, offering critical insights into the trade-offs between these objectives.

NSGA-II was initiated with a population size of 50, ensuring sufficient diversity for comprehensive exploration of the solution space. The Pareto front, depicted in Fig. 12, represents the set of non-dominated solutions where further reduction in $Error_{G_s}$ and $Error_{\eta}$ is not possible without compromising the other.

NSGA-II demonstrated significant advancements in the initial generations, with the Pareto front stabilizing after approximately 4,800 generations. The solutions along the Pareto front underline the balance between minimizing $Error_{G_s}$ and $Error_{\eta}$, providing a range of optimal trade-offs for informed decision-making. The Pareto front generated by NSGA-II is illustrated in Fig. 13, with each point representing a unique solution where further reduction in one error metric $Error_{G_s}$ and $Error_{\eta}$ results in an increase in the other. This multi-objective optimization approach offers valuable insights into the trade-offs between these conflicting objectives. Engineers can utilize the Pareto front to select solutions tailored to application-specific needs, such as prioritizing energy dissipation (η) or stiffness (G_s). NSGA-II demonstrated robust convergence behavior, progressively improving solution quality over successive generations. During the initial stages, the algorithm emphasized global exploration to uncover diverse solutions across the search space. In later stages, the crowding distance mechanism refined the population, enhancing both the quality and diversity of solutions on the Pareto front. By the 4,800th generation, the Pareto front exhibited minimal changes, signifying convergence, and providing a stable set of optimal trade-offs.

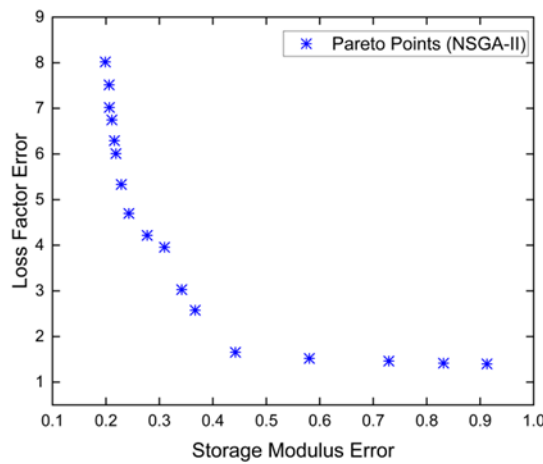


Fig. 12. Pareto optimal points by NSGA-II

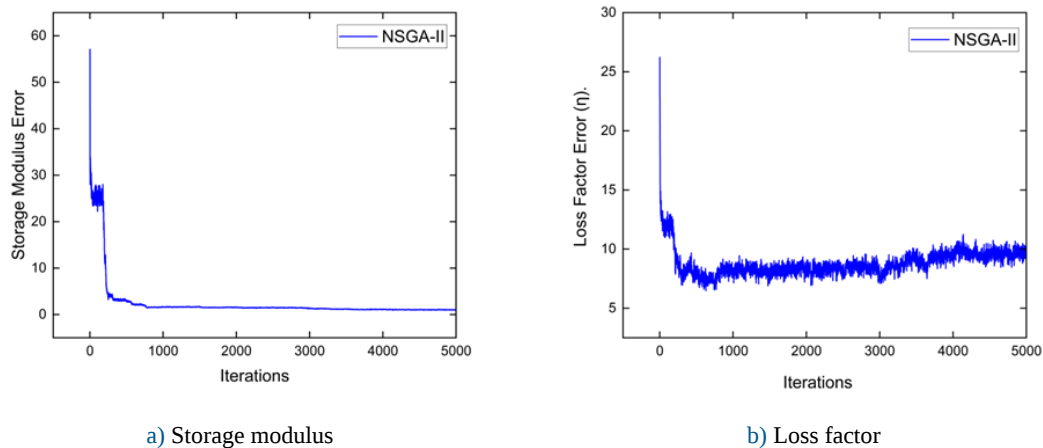


Fig. 13. Iterations vs. average error NSGA2

The results for storage modulus show in Fig. 13(a) indicate that the error achieved by the NSGA-II reached 2.6% after 5000 generations for the storage modulus. And reached 2.9 % after 5000 generations as shown in the Fig. 13(b) for the loss factor modulus. However, the plot reveals minor oscillations in the error values, suggesting that the optimization did not reach complete stability throughout the optimization process. This observation highlights the need for careful evaluation of the optimization trajectory and underscores the potential for further refinement in future iterations.

11. Conclusions

- This study effectively optimized the parameters of the modified fractional derivative constitutive model (MFDCM). for viscoelastic dampers (VEDs) by integrating multi-objective and single-objective optimization strategies. Using Non-dominated Sorting Genetic Algorithm II (NSGA-II), Teaching–Learning-Based Optimization (TLBO), Particle Swarm Optimization (PSO), and Harmony Search Algorithm (HSA), the research showcased the efficiency of these methods in reducing errors in both storage modulus (G_s) and loss factor (η), thereby enhancing the model's accuracy and reliability.
- The optimization problem was approached as a multi-objective challenge, addressing the conflicting goals of minimizing errors in storage modulus (G_s) and loss factor (η). Non-dominated Sorting Genetic Algorithm II (NSGA-II) was employed to construct a Pareto front, which provided valuable insights into the trade-offs between these objectives. This analysis guided the selection of weights ($W_{G_s} = 1, W_{\eta} = 5.5$) to transform the multi-objective problem into a single-objective framework.
- The single-objective optimization techniques then utilized these weights to fine-tune the parameters with precision, achieving a robust balance between computational efficiency and model fidelity.
- Non-dominated Sorting Genetic Algorithm II (NSGA-II) generated a comprehensive Pareto front, offering tailored solutions that cater to specific application requirements. The trade-offs illustrated by the Pareto front provided valuable insights into the interaction between storage modulus (G_s) and loss factor (η), facilitating informed weight selection for optimization. The Pareto front reached stability after approximately 4,800 generations, demonstrating the robustness and reliability of NSGA-II in addressing conflicting objectives effectively
- Harmonic Search Algorithm (HSA) effectively reduced errors, achieving a final error rate of 2.1% after 16,136 iterations. The algorithm maintained a balanced strategy between exploration and exploitation, ensuring both computational efficiency and high accuracy in the optimization results.
- Particle Swarm Optimization (PSO) demonstrated superior convergence speed compared to the other algorithms, stabilizing at an error rate of 1.92% within just 191 iterations. Its adaptive strategy for balancing global exploration and local exploitation proved highly effective, especially in optimization scenarios constrained by computational resources.
- Teaching–Learning-Based Optimization (TLBO) reached a stable error rate of 2.4% after 8,597 generations. Its structured approach to exploration and refinement facilitated robust convergence to near-optimal solutions, highlighting its dependability in addressing complex multi-parameter optimization challenges.
- The transformation from a multi-objective to a single-objective optimization problem was accomplished by selecting weights derived from the Non-dominated Sorting Genetic Algorithm II (NSGA-II) Pareto front. The weights ($W_{G_s} = 1, W_{\eta} = 5.5$) were optimized to balance the errors in storage modulus (G_s) and loss factor (η), effectively. Additionally, parameter boundaries were carefully established within physically meaningful ranges, ensuring both the validity and reliability of the optimization process and its outcomes.

Conflict of interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

This research received no external funding.

Data availability statement

Data generated during the current study are available from the corresponding author upon reasonable request.

References

- [1] Rashed NA, Ali YH, Rashid TA, Salih A (2024) Unraveling the versatility and impact of multi-objective optimization: Algorithms, applications, and trends for solving complex real-world problems. *Int J Multi-Obj Opt* 15(3):275–295. <https://doi.org/10.1234/j.moo.2024.03.001>
- [2] Padamwar BV, Pandey H (2019) Optimization techniques in operations research: A review. *Turk J Comput Math Educ* 10(1):746–752. <https://doi.org/10.61841/turcomat.v10i1.14604>
- [3] Gad AG (2022) Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. *Arch Comput Methods Eng* 29:2531–2561. <https://doi.org/10.1007/s11831-021-09694-4>
- [4] Selvarajan S (2024) A comprehensive study on modern optimization techniques for engineering applications. *Artif Intell Rev* 57:194. <https://doi.org/10.1007/s10462-024-10829-9>
- [5] Pereira JLJ, Oliver GA, Francisco MB, Cunha SS Jr, Gomes GF (2022) A review of multi-objective optimization: Methods and algorithms in mechanical engineering problems. *Arch Comput Methods Eng* 29:2285–2308. <https://doi.org/10.1007/s11831-021-09663-x>
- [6] Wei Y, Peng B, Xie R, Chen Y, Qin Y, Wen P, Bauer S, Tung P-Y (2024) Derivative-free tree optimization for complex systems. *JOM* 73:1–39. <https://doi.org/10.2404.04062v17>
- [7] Bhourri MA, Joly M, Yu R, Sarkar S, Perdikaris P (2023) Scalable Bayesian optimization with high-dimensional outputs using randomized prior networks. Preprint, arXiv:2302.07260v5. <https://doi.org/10.48550/arXiv.2302.07260>
- [8] Zhang T, Xu ZD, Huang XH, Dong YR, Shi QX (2023) Study on the modified fractional derivative constitutive model of viscoelastic dampers. *International Journal of Non-Linear Mechanics* 155:104462. <https://doi.org/10.1016/j.ijnonlinmec.2023.104462>
- [9] Liu X, Qin J, Zhao K, Featherston CA, Kennedy D, Jing Y, Yang G (2023) Design optimization of laminated composite structures using artificial neural network and genetic algorithm. *Composite Structures* 305:116500. <https://doi.org/10.1016/j.compstruct.2022.116500>
- [10] Keshtegar B, Nguyen-Thoi T, Truong TT, Zhu SP (2021) Optimization of buckling load for laminated composite plates using adaptive Kriging-improved PSO: A novel hybrid intelligent method. *Defence Technol* 17:85–99. <https://doi.org/10.1016/j.dt.2020.02.020>
- [11] Nguyen H, Moayedi H, Foong LK, Najjar HA, Wan Jusoh WA, Rashid AS, Jamali J (2020) Optimizing ANN models with PSO for predicting short building seismic response. *Eng Comput* 36:823–837. <https://doi.org/10.1007/s00366-019-00733-0>
- [12] Minh HL, Khatir S, Abdel Wahab M, Le TC (2021) An Enhancing Particle Swarm Optimization Algorithm (EHVPSO) for damage identification in 3D transmission tower. *Eng Struct* 242:112412. <https://doi.org/10.1016/j.engstruct.2021.112412>
- [13] Wu J, Cheng F, Zou C, Zhang R, Li C, Huang S, Zhou Y (2022) Swarm intelligent optimization conjunction with Kriging model for bridge structure finite element model updating. *Buildings* 12:504. <https://doi.org/10.3390/buildings12050504>
- [14] Altun S, Varjovi MH, Karci A (2018) Performance comparison of different optimization methods under the same conditions. *IEEE, Malatya, Türkiye*. <https://doi.org/10.1109/IDAP.2018.8620761>

- [15] Jain A, Lalwani S, Lalwani M (2019) A comparative analysis of MOPSO, NSGA-II, SPEA2, and PESA2 for multi-objective optimal power flow. *Int J Eng Res Technol* 8(3):234–240. <https://doi.org/10.5281/zenodo.331606195>
- [16] Wen X, Song Q, Qian Y, Qiao D, Wang H, Zhang Y, Li H (2023) Effective Improved NSGA-II Algorithm for Multi-Objective Integrated Process Planning and Scheduling. *Mathematics* 11:3523. <https://doi.org/10.3390/math11163523>
- [17] Jain A, Lalwani S, Lalwani M (2018) A comparative analysis of MOPSO, NSGA-II, SPEA2 and PESA2 for multi-objective optimal power flow. *IEEE International Conference on Emerging Technologies*. <https://doi.org/10.1109/EPETSG.2018.8659054>
- [18] Yilun L, Zhengwei X, Shiyu Y, Zhuoxiang R (2022) A hybrid algorithm based on NSGA-II and MOPSO for multi-objective designs of electromagnetic devices. *IEEE 20th Biennial Conference on Electromagnetic Field Computation (CEFC)*. <https://doi.org/10.1109/CEFC55061.2022.9940652>
- [19] Cao Y, Su F, Antwi-Afari MF, Lei J, Wu X, Liu Y (2024) Enhancing mix proportion design of low-carbon concrete for shield segment using a combination of Bayesian optimization-NGBoost and NSGA-III algorithm. *J Clean Prod* 465:142746. <https://doi.org/10.1016/j.jclepro.2024.14274621>
- [20] Li Y, Li Y, Liu J, Li W, She L (2024) Optimization of some properties of hydraulic asphalt concrete mix using RSM and NSGA-II. *Constr Build Mater* 438:137317. <https://doi.org/10.1016/j.conbuildmat.2024.137317>
- [21] Fan Y, Xin J, Yang L, Zhou J, Luo C, Zhou Y, Zhang H (2024) Optimization method for the length of the outsourcing concrete working plane on the main arch rib of a rigid-frame arch bridge based on the NSGA-II algorithm. *Structures* 59:105767. <https://doi.org/10.1016/j.istruc.2023.105767>
- [22] Zhang W, Jiang S, Li X, Chen Z, Cao G, Mei M (2024) Multi-objective optimization of concrete pumping S-pipe based on DEM and NSGA-II algorithm. *Powder Technol* 434:119314. <https://doi.org/10.1016/j.powtec.2023.119314>
- [23] Thai MV, Galimard P, Elachachi SM, Ménard S (2022) Multi-objective optimization of cross-laminated timber-concrete composite floor using NSGA-II. *J Build Eng* 52:104285. <https://doi.org/10.1016/j.jobte.2022.104285>
- [24] Jin L, Zhang Y, Liu P, Fan T, Wu T, Wu Q (2024) Carbon-footprint based concrete proportion design using LSTM and MOPSO algorithms. *Mater Today Commun* 38:107837. <https://doi.org/10.1016/j.mtcomm.2023.107837>
- [25] Safari D, Maheri MR, Maheri A (2011) Optimum design of steel frames using a multiple-deme GA with improved reproduction operators. *J Constr Steel Res* 67:1232–1243
- [26] Qiao S, Dai X, Liu Z, Huang J, Zhu G (2012) Improving the optimization performance of NSGA-II algorithm by experiment design methods. In: *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 2713–2719. <https://doi.org/10.1109/CEC.2012.6269589>
- [27] Ma H, Zhang Y, Sun S, Liu T, Shan Y (2023) A comprehensive survey on NSGA-II for multi-objective optimization and applications. *Artif Intell Rev* 56:15217–15270. <https://doi.org/10.1007/s10462-023-10526-z>
- [28] Pritz T (2003) Five-parameter fractional derivative model for polymeric damping materials. *J. Sound Vib* 265:935–952. [https://doi.org/10.1016/S0022-460X\(02\)01530-4](https://doi.org/10.1016/S0022-460X(02)01530-4)
- [29] Xu Y, Xu Z, Guo Y, Ge T, Xu C, Huang X (2019) A theoretical and experimental study of viscoelastic damper based on fractional derivative approach and micromolecular structures. *J. Vib. Acoust. Trans. ASME*. 141(3):031010. <https://doi:10.1115/1.4042517>
- [30] Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197. <https://doi.org/10.1109/4235.996017>
- [31] Coello Coello CA, Lamont GB, Van Veldhuizen DA (2007) *Evolutionary Algorithms for Solving Multi-Objective Problems*, Second Edition. Springer Science+Business Media, New York. <https://doi.org/10.1007/978-0-387-36797-2>
- [32] Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Comput Aided Des* 43:303–315. <https://doi.org/10.1016/j.cad.2010.12.015>
- [33] Das I, Dennis JE (1997) A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Struct Multidiscip Optim* 14:63–69. <https://doi.org/10.1007/BF01197559>
- [34] Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*. IEEE, Perth, Australia. <https://doi.org/10.1109/ICNN.1995.488968>

- [35] Yang XS (2010) Engineering Optimization: An Introduction with Metaheuristic Applications. Wiley, Cambridge. <https://doi.org/10.1002/9781118687441>
- [36] Deb K, Jain H, Kumar A (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2):182–197. <https://doi.org/10.1109/TEVC.2002.1009193>
- [37] Hojjati A, Monadi M, Faridhosseini A, Mohammadi M (2018) Application and comparison of NSGA-II and MOPSO in multi-objective optimization of water resources systems. *Journal of Hydrology and Hydromechanics* 66(3):323–329. <https://doi.org/10.2478/johh-2018-0006>
- [38] Vo-Duy T, Duong-Gia D, Ho-Huu V, Vu-Do HC, Nguyen-Thoi T (2017) Multi-objective optimization of laminated composite beam structures using NSGA-II algorithm. *Composites Structures* 168:498–509. <https://doi.org/10.1016/j.compstruct.2017.02.030>
- [39] Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: Harmony Search. *Simulation* 76(2):60–68. [https://doi.org/10.1016/S0037-5497\(01\)00125-6](https://doi.org/10.1016/S0037-5497(01)00125-6)
- [40] Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: Harmony Search. *Simulation* 76(2):60–68. [https://doi.org/10.1016/S0037-5497\(01\)00125-6](https://doi.org/10.1016/S0037-5497(01)00125-6)
- [41] Qin F, Mohd Zain A, Zhou KQ (2022) Harmony search algorithm and related variants: A systematic review. *Swarm and Evolutionary Computation* 74:101126. <https://doi.org/10.1016/j.swevo.2022.101126>
- [42] Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design* 43(3):303–315. <https://doi.org/10.1016/j.cad.2010.12.015>
- [43] Kennedy J, Eberhart RC (1995) Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*. <https://doi.org/10.1109/ICNN.1995.488968>
- [44] Genovesi S, Mittra R, Monorchio A, Manara G (2006) Particle Swarm Optimization for the Design of Frequency Selective Surfaces. *IEEE Antennas and Wireless Propagation Letters* 5:277–280. <https://doi.org/10.1109/LAWP.2006.879703>
- [45] Ramírez-Ochoa DD, Pérez-Domínguez LA, Martínez-Gómez EA, Luviano-Cruz D (2022) PSO, a swarm intelligence-based evolutionary algorithm as a decision-making strategy: A review. *Symmetry* 14(3):455. <https://doi.org/10.3390/sym14030455>