

Article

Probabilistic Structural Model Updating with Modal Flexibility Using a Modified Firefly Algorithm

Zhouquan Feng ^{1,2} , Wenzan Wang ¹ and Jiren Zhang ^{1,*} 

¹ Key Laboratory of Wind and Bridge Engineering of Hunan Province, College of Civil Engineering, Hunan University, Changsha 410082, China

² Research Institute of Hunan University in Chongqing, Chongqing 401151, China

* Correspondence: jrzhang1994@hnu.edu.cn

Abstract: Structural model updating is one of the most important steps in structural health monitoring, which can achieve high-precision matching between finite element models and actual engineering structures. In this study, a Bayesian model updating method with modal flexibility was presented, where a modified heuristic optimization algorithm named modified Nelder–Mead firefly algorithm (m-NMFA) was proposed to find the most probable values (MPV) of model parameters for the maximum a posteriori probability (MAP) estimate. The proposed m-NMFA was compared to the original firefly algorithm (FA), the genetic algorithm (GA), and the particle swarm algorithm (PSO) through the numerical illustrative examples of 18 benchmark functions and a twelve-story shear frame model. Then, a six-story shear frame model test was performed to identify the inter-story stiffness of the structure in the original and the damage states, respectively. By comparing the two, the position and extent of damage were accurately found and quantified in a probabilistic manner. In terms of optimization, the proposed m-NMFA was powerful to find the MPVs much faster and more accurately. In the incomplete measurement case, only the m-NMFA achieved target damage identification results. The proposed Bayesian model updating method has the advantages of high precision, fast convergence, and strong robustness in MPV finding and the ability of parameter uncertainty quantification.

Keywords: structural health monitoring; structural model updating; modified firefly algorithm; modal flexibility; damage detection



Citation: Feng, Z.; Wang, W.; Zhang, J. Probabilistic Structural Model Updating with Modal Flexibility Using a Modified Firefly Algorithm. *Materials* **2022**, *15*, 8630. <https://doi.org/10.3390/ma15238630>

Academic Editors: Rafal Burdzik, Maosen Cao, Mahmoud Bayat, Wei Xu and Nizar Faisal Alkayem

Received: 14 November 2022

Accepted: 28 November 2022

Published: 3 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Structural model updating is an important part of structural health monitoring. Using monitoring data, structural models can be updated to more realistically represent the actual state of the structures and to obtain mechanical analytical results which are closest to the real state of the structures. In fact, structural model updating is an inverse problem of structural analysis, which can be regarded as an optimization problem. Since the 1960s, the field of structural model updating has developed a variety of methods. However, errors exist widely in experimental testing and modeling which are caused by various factors [1,2]. Shinozuka [3] and Collins et al. [4] introduced an uncertainty concept and used a probability statistical method to evaluate the uncertainty of parameters while updating the model parameters. As a result, structural model updating has had a milestone leap.

Probabilistic model updating methods can be roughly divided into stochastic finite element corrections and finite element corrections based on Bayesian methods. Among them, stochastic finite element correction is a large category, which includes model correction based on the Monte Carlo method and model correction based on the perturbation method. Beck and Katafygiotis et al. [5,6] first introduced the Bayesian method into model updating in 1998 and established a preliminary theoretical framework; they then used the asymptotic approximation method to estimate a posterior probability in the parameter

space. Researchers including Beck, Katafygiotis, Yuen, and Feng have conducted a series of in-depth studies on the problems of incomplete measurement, complex likelihood function, and posterior probability that are difficult to solve, and proposed "system modal shape", modal flexibility method, local observation concepts and methods, such as data fusion, to further enrich the relevant theories [7–12].

Based on the Bayesian model updating framework, model parameters are embedded in the posteriori probability function. Generally, there are two kinds of methods to solve the maximum a posteriori probability (MAP) estimate, one is the Laplace approximation and the other is random sampling of the posteriori probability distribution function. For a globally identifiable case with a large number of data, we can approximate the posterior probability distribution function as a Gaussian distribution including mean and variance, which mean is the most probable value (MPV), and the obtained MPV is the MAP estimate for the model parameters. Because MPV takes the maximum function value for the posterior probability distribution function, the solution of MPV eventually becomes an optimization problem. After completing the construction of a mathematical model for the optimization problem, the optimization algorithm can be used to find the optimal value of the mathematical model. Due to the limitations of analytical methods, numerical methods have become the most widely used methods for solving optimization problems. They can be mainly divided into two categories. The first type includes traditional optimization algorithms focusing on single-objective optimization problems, such as the Newton Iteration Method (NIM), the Conjugate Direction Method (CDM), etc. The other type includes Heuristic Algorithms (HA) for multi-objective optimization problems. Inspired by bionics, scholars have developed Metaheuristic Algorithms (MA) [13,14]. A meta-heuristic optimization algorithm allows randomness in the optimization process of the algorithm by introducing a mutation factor, thereby reducing the possibility of falling into a local optimum. At present, scholars have proposed many meta-heuristic optimization algorithms, such as the Genetic Algorithm (GA) [15], the Ant Colony Optimization (ACO) [16], the Firefly Algorithm (FA) [17], etc.

The firefly algorithm was first proposed by Yang in 2007. It is a global optimization algorithm inspired by the glowing behavior of fireflies in nature. Since the firefly algorithm was proposed, due to its simple parameters, fast convergence speed, high precision, as well as its own automatic population segmentation ability and nonlinear individual attraction mechanism, it has received extensive attention from the academic community. However, like other swarm intelligence optimization algorithms, this algorithm is prone to problems such as premature maturity, falling into local optimum, and insufficient ability to solve multi-dimensional optimization problems. Therefore, it needs to be appropriately improved when solving actual complex optimization problems.

Based on the original FA algorithm, this paper makes several modifications and introduces the Nelder–Mead Algorithm (NMA) for local search and proposes a modified Nelder–Mead firefly algorithm. Three improvements are proposed: modification in the control parameters, introduction of a boundary constrain mechanism, and introduction of a diversity threshold. Eighteen benchmark functions and a 12-story shear frame model are used to test the performance of the m-NMFA and to provide support for the comparison of the original FA, the GA, and the PSO to the m-NMFA. A six-story steel shear frame was designed to perform a vibration experiment for verifying the effectiveness of the m-NMFA in Bayesian structural model updating.

2. Model Updating Formulation

2.1. Modal Flexibility

The stiffness matrix K and flexibility matrix C of a linear dynamical system with N degrees of freedom can be written as follows [18]:

$$K = M\Phi\Lambda\Phi^T M = M\left(\sum_{i=1}^N \omega_i^2 \phi_i \phi_i^T\right)M. \quad (1)$$

$$C = \Phi \Lambda^{-1} \Phi^T = \sum_{i=1}^N \frac{1}{\omega_i^2} \phi_i \phi_i^T \quad (2)$$

where M and K are the mass matrix and the stiffness matrix, respectively; $\Phi = [\phi_1, \phi_2, \dots, \phi_n]$ is the mass-normalized mode shape matrix; and Λ is the diagonal eigenvalue with squared natural circular modal frequencies.

In practice, it is always the case that the spatial arrangement of sensors attached to the structure is generally limited and only a limited number of DOFs can be observed, for which only incomplete spatial mode shapes can be gained. It can be seen from Equation (2) that the greater the modal frequency is, the smaller the contribution of one mode to the flexibility matrix C will be, which is on the contrary to that of Equation (1) and is more practical in solving vibration-related problems. Therefore, the flexibility matrix C can be established approximately with only low-frequency modes of interest. Therefore, the Equation (2) yield as follows [19]:

$$C^m \approx C_t^m = \sum_{i=1}^{N_m} \frac{1}{\omega_i^2} \phi_i^m (\phi_i^m)^T \quad (3)$$

where C^m denotes the flexibility submatrix with respect to the measured DOFs; C_t^m denotes the truncated flexibility submatrix; N_m denotes the number of selected low-frequency modes (frequency from low to high and no interval in this research); and $\phi_i^m \in R^{N_o \times 1}$ ($i = 1, 2, \dots, N_m$) denotes the incomplete mode shapes. Needless to say, C_t^m is supposed to be as close as possible to C^m ; if all the DOFs of a structure are well observed and all the mode shapes are gained, C_t^m will be exactly equal to C^m .

Mass-normalization is a key issue to build up the flexibility matrix. If the measurement is incomplete, the dimension of modal parameters, such as mode shapes, would be less than that of the mass matrix M . To handle this problem, the SEREP method [20] is introduced to convert M to reduced mass matrix M^m . Then, the incomplete mode shape matrixes are normalized as follows:

$$(\Phi^m)^T M^m \Phi^m = I \quad (4)$$

2.2. Model Reduction

System equivalent reduction and expansion process (SEREP) is an eigenvalue-based reduction technique used to approximate the real structural state vectors with mode shape measured from the DOFs of the user's interest [20].

In the method of SEREP, the first step is to separate the mass and stiffness matrices as follows:

$$M^P = \begin{bmatrix} M_{mm} & M_{mu} \\ M_{um} & M_{uu} \end{bmatrix}, \quad K^P = \begin{bmatrix} K_{mm} & K_{mu} \\ K_{um} & K_{uu} \end{bmatrix}. \quad (5)$$

where m and u denote the measured and unmeasured DOFs. Then, the SEREP transformation matrix can be written as follows:

$$T = \begin{bmatrix} \Phi^m \\ \Phi^u \end{bmatrix} (\Phi^m)^*. \quad (6)$$

where Φ^m denotes the retained mode shape matrix with N_o (the number of observed DOFs) rows and N_m columns, and $(\Phi^m)^*$ represents the generalized inverse of matrix Φ^m . The process of converting Φ^m to $(\Phi^m)^*$ is shown in [11,20,21]. Ultimately, the reduced mass and stiffness matrices can be written as follows:

$$M^m = T^T M^P T, \quad K^m = T^T K^P T. \quad (7)$$

2.3. Bayesian Model Updating Based on Measured Modal Flexibility Data

Bayesian method for finite element model updating is one of the most commonly used probabilistic methods for structural model updating, and it has been confirmed to

be efficient and useful [22,23]. In this section, modal flexibility data are used to establish a posterior-PDF-based objective function.

As the modal flexibility matrix is symmetrical, we can use the lower triangular matrix of the flexibility matrix C^l to establish the objective function. Let f denote the modal flexibility data, and the matrix C^l is vectorized based on the following equation:

$$f = (c_{11}, c_{21}, \dots, c_{N_0 1}, \dots, c_{ii}, c_{(i+1)i}, \dots, c_{N_0 i}, \dots, c_{N_0 N_0}) \quad (8)$$

where c denotes the element in the matrix C^l .

It is supposed that the test is repeated N_t times and D denotes the experimental data. Then, the matrix D can be written as follows:

$$D = \{ \hat{f}_1, \hat{f}_2, \dots, \hat{f}_{N_t} \} \quad (9)$$

As the modal can be described by the probability distributions of both the unknown parameters and the prediction error [5], the measured data can be written as follows:

$$\hat{f}_t = f(\theta) + e_t \quad (10)$$

where $f(\theta)$ denotes the prediction of the analytical model and is defined as the vector for the given values of the model parameters θ . The term e_t denotes the prediction error and is assumed to follow the maximum entropy principle [17]. That is to say, e_t yields a multi-dimensional Gaussian distribution with zero mean and covariance matrix Σ [5]. The covariance matrix is computed based on the modal flexibility vectors as follows:

$$\Sigma = \frac{1}{N_t - 1} \sum_{t=1}^{N_t} (\hat{f}_t - \bar{f}) (\hat{f}_t - \bar{f})^T \quad (11)$$

where \bar{f} contains the average value of N_t experimental modal flexibility vectors. It can be expressed using the following equation:

$$\bar{f} = \frac{1}{N_t} \sum_{t=1}^{N_t} \hat{f}_t \quad (12)$$

Then, the likelihood function of the t -th set of modal flexibility vector data can be expressed as follows:

$$p(\hat{f}_t | \theta, M) = C_1 \exp\left(-\frac{1}{2} (\hat{f}_t - f(\theta))^T \Sigma^{-1} (\hat{f}_t - f(\theta))\right) \quad (13)$$

where C_1 is a constant. Assuming that the N_t sets of the modal flexibility vector datas are mutually independent, the likelihood function of D can be written as follows:

$$p(D | \theta, M) = C_2 \exp\left(-\frac{1}{2} \sum_{t=1}^{N_t} (\hat{f}_t - f(\theta))^T \Sigma^{-1} (\hat{f}_t - f(\theta))\right) \quad (14)$$

where C_2 is also a constant like C_1 . As the prior PDF $p(\theta)$ follows a non-informative uniform distribution, the posterior PDF $p(\theta | D, M)$ can be written in a similar form as the likelihood function in Equation (14). Given that the observed DOFs are limited, the SEREP method will be utilized here and then the measured mode shapes are mass-normalized with respect to the reduced model mass M^m . It is easy to see that M^m depends on θ only and so does the posterior PDF $p(\theta | D, M)$. On this basis, the most probable stiffness parameter vector $\hat{\theta}$ can be gained by maximizing the posterior PDF value or minimizing its negative

value. Here, the objective function is defined as a truncated part of Equation (14) and is simplified with the following logarithm:

$$J(\theta) = -\ln p(\theta|D, M) \quad (15)$$

where $J(\theta)$ denotes the objective function. Substitute Equation (14) into Equation (15), the function is expressed as follows:

$$J(\theta) = \frac{1}{2} \sum_{t=1}^{N_t} (\hat{f}_t - f(\theta))^T \Sigma^{-1} (\hat{f}_t - f(\theta)) \quad (16)$$

Till now, the objective function has been established and the model updating problem has been transformed as an optimization problem. When the most probable stiffness parameter vector $\hat{\theta}$ is found during the process of minimizing $J(\theta)$, the actual stiffness matrix of a structure is ensured naturally.

The posterior PDF can be well approximated using a set of multi-dimensional Gaussian distribution with $\hat{\theta}$ and Σ_{θ} as its mean value and covariance matrix.

$$\Sigma_{\theta} = [H(\hat{\theta})]^{-1} = [\nabla J(\hat{\theta}) \nabla^T \nabla J(\hat{\theta}) \nabla^T]^{-1} \quad (17)$$

where $H(\hat{\theta})$ is the Hessian matrix of objective function.

In order to search for the best-fit vector $\hat{\theta}$ exactly and reliably, a modified FA algorithm is proposed, which is introduced in the next section and utilized in the numerical and experimental examples in Sections 4 and 5.

3. Firefly Algorithm

3.1. Original Firefly Algorithm

Firefly Algorithm (FA) was first proposed by Xin-She Yang at Cambridge University in 2007 [24] and has become a widely used global optimization algorithm. In the original FA, it is not gender difference but the intensity of light I , which decreases with the increase in the square of Cartesian distance between two fireflies r^2 , that distinguishes fireflies. Fireflies that have greater I values are regarded as fitting the solution to the objective function better, and they are supposed to attract the ones with smaller values to them. Given that light will gradually weaken in a medium as distance increases, the attractiveness parameter β , which denotes the attraction between any two fireflies, is introduced in this algorithm. Similar to the intensity of light, the greater the distance is, the smaller the attractiveness parameter β will be, and the attraction between two fireflies will become weaker. Furthermore, the light absorption parameter γ is introduced to denote the ability of a medium to absorb light, which is proposed to simulate the natural environment that fireflies are in.

The behavior of firefly is idealized for algorithmic use, and three rules are introduced as follows [24]:

- All the fireflies are unisex so they can attract each other without the influence of gender.
- The attractiveness is proportional to the brightness, and they both decrease as their distance increases.
- The brightness of a firefly is determined by the landscape of the objective function.

In a given medium, light intensity I decreases in a Gaussian form as follows:

$$I(r) = I_0 e^{-\gamma r^2} \quad (18)$$

where I_0 represents the intensity at $r = 0$, and r denotes the distance between any two fireflies. It can be seen that light intensity decreases exponentially and the influence of the absorption coefficient γ on light intensity is defined in the exponential term of the function mentioned above. However, light intensity is flexible in both definition and form since it is allowed to be the fitness value of a certain objective function to a practical problem [17].

The attractiveness β of a firefly is proportional to the light intensity I and is also defined in an exponential form as follows:

$$\beta = \beta_0 e^{-\gamma r^2} \quad (19)$$

where β_0 is the attractiveness value at $r = 0$.

The distance r between any adjacent fireflies x_i and x_j is expressed as follows:

$$r = \|x_i - x_j\| = \sqrt[2]{\sum_{k=1}^n (x_{i,k} - x_{j,k})^2} \quad (20)$$

where $x_{i,k}$ is the k -th component of the spatial coordinate x_i of i -th firefly, and n denotes the dimension of the objective function or problems to be optimized.

As mentioned above, fireflies with a greater value of light intensity I is supposed to attract the ones with a smaller value of I . Assume that the i -th firefly is weaker than the j -th, then the movement of firefly i to firefly j is described as follows:

$$x_i = x_i + \beta_0 e^{-\gamma r^2} (x_j - x_i) + \alpha \epsilon_i \quad (21)$$

where the second term is the attraction part, and the third term is the randomization part with step parameter α . ϵ_i is drawn from different distributions, such as Gaussian distribution, uniform distribution, and levy flight. In many cases, a uniform distribution is utilized as the simplest form and, thus, ϵ_i becomes a vector of random numbers uniformly generated in [17,25–27]. An example of a levy flight distribution is shown in [24].

The firefly algorithm is an efficient algorithm, and Algorithm 1's pseudocode is shown as follows:

Algorithm 1 the firefly algorithm

```

Initialize the parameters( $\alpha, \beta, \gamma, n$ )
Initialize randomly a population of  $n$  fireflies
Evaluate the fitness of the initial population at  $x_i$  by objective function
  While ( $k < \text{MaxGen}$ ) do
    For  $i = 1:n$ 
      For  $j = 1:n$ 
        If Firefly  $j$  is better than  $i$ 
          Firefly  $i$  moves towards  $j$ 
        End if
      End for
      Evaluate the new solution
    End for
    Rank and update the best solution found so far
    Update iteration counter  $k$ ;
    Update  $\alpha$ 
  End while

```

3.2. Modified Nelder–Mead Firefly Algorithm

The modified Nelder–Mead firefly algorithm(m-NMFA) is generally a hybrid version combining the advantages of the local optimization algorithm, the Nelder–Mead algorithm (NMA), and the global optimization algorithm. In order to strengthen the superiority of the hybrid algorithm, three improvements are proposed: modification in the parameters α and β , introduction of a boundary constrain mechanism, and introduction of a diversity threshold.

3.2.1. Modification in Parameters α and β

Multiple studies have proved that the parameter α and the attractiveness parameter β are of great influence in both convergence speed and accuracy of results [24,24–33].

Yang in 2008 [17] used a fixed-step parameter $\alpha = 0.2$ to solve a four-peak function and then pointed out that a decreasing α could improve the convergence of the FA. Two examples are given as follows:

$$\alpha = \alpha_{\infty} + (\alpha_0 - \alpha_{\infty})e^{-t} \quad (22)$$

and

$$\alpha = \alpha_0\theta^t \quad (23)$$

where α_0 is the initial step parameter and α_{∞} is the final value. $\theta \in [0,1]$ is the randomness reduction constant. α iterates as the iteration counter t increases. It can be seen that the step parameter α is limited between α_0 and α_{∞} in Formula (22) and gradually approaches α_{∞} . Similarly, α gradually approaches zero in Equation (23). Moreover, in Equation (23), the θ and α_0 values have great influence in the decay speed of step parameter α in the process of iteration, which will produce a unknown effect—positive or negative—on the solution result. In previous studies, different θ values are taken to cope with different problems. Yang and He [34] pointed out that θ could take values from 0.95 to 0.97, Manoharan [30] and Rizk-Allah [35] took θ as 0.9, and Wang [31] took θ as 0.7 and α_0 as 0.25. Moreover, a noticeable issue has been highlighted that the scaling parameter S_d should be introduced in consideration of the actual scales of the problem of interest [36].

$$S_d = u_d - l_d \quad (24)$$

where d denotes the d -th dimension unknown variable, and u_d and l_d are the upper and lower limit values. Therefore, ϵ_i in Equation (21) should be replaced by $\epsilon_i S$, where $S = \{S_d, d=1,2, \dots, n\}$.

Beside the strategies mentioned above, various versions of the step parameter iteration formula have been proposed in the last decade. Brajević [27] proposed a novel strategy that iterates α with the maximum number of interaction and the initial step parameter α_0 and it is written as follows:

$$\alpha_{\kappa}^{(t)} = \alpha_{\kappa}^{(t-1)} \left(\frac{1}{90,000} \right)^{\frac{2}{MaxGen}} \quad (25)$$

where κ is the identifier of each firefly and takes the dimension of the problem as its maximum value; $MaxGen$ denotes the maximum number of interaction; and t denotes the current number of iterations. The step parameter decreases more quickly and will be much smaller than that in Equation (23). Ignoring the distinction among all the fireflies, Equation (25) can be expressed by another form as follows:

$$\alpha(t) = \alpha_0 \Theta^{\frac{\rho t}{MaxGen}} \quad (26)$$

where $\Theta = (1/90,000)^2 \approx 1.234 \times 10^{-10}$, and ρ is defined as the reference coefficient and its value is 1. Previous tests have shown that the results drawn from an α that is of a slightly smaller value during the previous iterations (take 50 iterations for example) tend to outperform the results drawn from a greater α value in convergence speed. Previous tests have also shown that the convergence speed of the algorithm increases but the accuracy decreases gradually if ρ is taken to be an integer constant value ranging from 1 to 3. That is to say, the decrease rate of α in a later course of iteration should be well controlled to have a good balance between exploration and exploitation to the maximum. So, let ρ be expressed in a nonlinear form using the following equation:

$$\rho = 1 + 2 \left(\frac{MaxGen - t}{MaxGen} \right)^2 \quad (27)$$

Figure 1 shows the decreasing process of the reference coefficient.

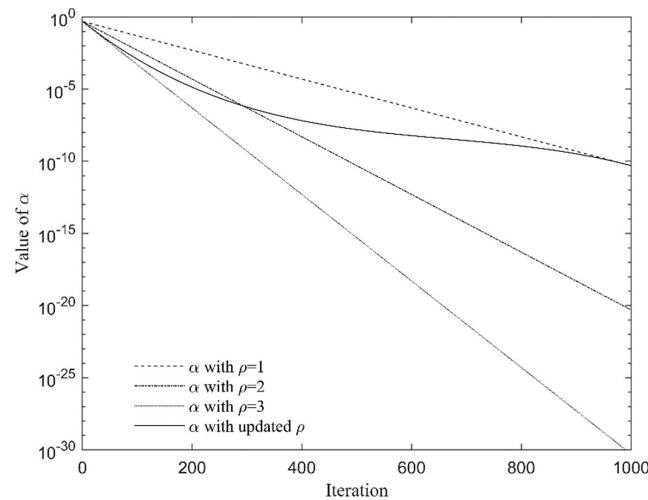


Figure 1. Iteration of the reference coefficient ($\Theta = 1.0 \times 10^{-10}$, $\alpha_0 = 0.5$).

Apart from the modification on the step parameter, extensive efforts have been made to modify the attraction parameter β and the light absorption parameter γ . If β_0 and γ are taken to be constants, the distance between any two adjacent fireflies will approach zero gradually. As the iteration number increases, it will ideally follow the equation below:

$$\lim_{t \rightarrow \infty} \beta = \lim_{t \rightarrow \infty} \beta_0 e^{-\gamma t^2} = \beta_0 \tag{28}$$

This ensures that the attraction within a firefly colony will always be a contributing factor. In practice, however, it would be a tough problem in high-dimensional problems if the distance between any two firefly individuals is too large at the very beginning of the iteration progress and the attraction is too weak to get the individuals closer. As a consequence, the algorithm stagnates. Moreover, the higher the dimension of a project is, the more easily this phenomenon will occur.

Selvarasu [33] proposed a modified β formula:

$$\beta = \beta_{min} + (\beta_{max} - \beta_{min})e^{-\gamma t^2} \tag{29}$$

where β_{min} and β_{max} are the lower and upper boundaries of the attraction parameter β . This strategy prevents the attraction from being excessively large or small. Herein, this paper utilizes Equation (29) to improve the performance of the FA in solving high-dimensional problems.

3.2.2. Boundary Constrain Mechanism

Similar to other swarm intelligence methods, such as the Bee Colony Algorithm (BCA), the Genetic Algorithm (GA), and the Particle Swarm Optimization (PSO), the FA will be ill-conditioned if the variables of firefly individuals exceed the feasible region. Yang introduced a boundary constrain mechanism that forces the outer variables to the border, and the formula is expressed as follows:

$$x_{i,j} = \begin{cases} l_j, & x_{i,j} \leq l_j \\ u_j, & x_{i,j} \geq u_j \end{cases} \tag{30}$$

where $x_{i,j}$ denotes the i -th firefly individual and the j -th variable, and l_j and u_j are the lower and upper boundaries of the j -th variable. On the one hand, this mechanism ensures that all the variables of one firefly are within the feasible range and, moreover, prevents the algorithm from falling into the local optimum or leading to a completely wrong solution.

On the other hand, it reduces the diversity of the firefly population to some extent. This mechanism has been utilized by Ivona Brajević [37] and Rajan [38].

The optimal solution of a multi-dimensional problem is determined by all the variables. It is possible that a newly generated firefly individual would deviate from the optimal solution with only several variables being updated. Therefore, a new boundary constraint mechanism is proposed. If any one of the variables overflows, a new firefly individual will be regenerated randomly, Algorithm 2 is:

Algorithm 2 boundary constrain mechanism

```

For i = 1:n
  For j = 1:D
    If  $x_{i,j} < l_j$  or  $x_{i,j} > u_j$ 
      Then  $x_{i,j} = l_j + \text{rand} \times (u_j - l_j)$ 
    End if
  End for
   $x_i = x_i + F^*(x_{\text{best}} - x_i)$ 
End for

```

where $\text{rand} \in [0, 1]$ denotes a random factor that is used to generate a new variable randomly, and $F \in [0, 0.5]$ is another random factor that leads the new firefly towards the current global best solution x_{best} . Both rand and F follow a uniform distribution.

3.2.3. Hybrid of Nelder–Mead Algorithm and the Diverse Threshold

The Nelder–Mead Algorithm is a local optimal algorithm that adapts itself to the local landscape and contracts on to the final minimum. This method for finding the local minimum is widely used and it can be realized expediently by using the built-in function “fminsearch” in MATLAB software. The way of combining the NMA and the FA is to start the NMA in the iterative process of the FA in order to dig out a better solution nearby the current global best solution found up to t -th ($t \leq \text{MaxGen}$) iteration.

However, the start point of the NMA in the iterative process needs to be well weighted because when to enable the NMA will greatly affect the solution accuracy and convergence speed. If the NMA starts too early, the current best solution will be far from reliable and the NMA will run meaninglessly, which will result in inefficiency. Meanwhile, if it starts too late, the contribution of the solution searched by the NMA to the algorithm will be weakened. Therefore, the concept of diversity threshold is introduced. When the diversity of a firefly population reaches a certain threshold, the NMA is enabled to balance the contradiction mentioned above.

Let $x_{k,\text{best}}$ and $x_{k,\text{worst}}$ denote the best and worst firefly individuals of the single k -th iteration, and $x_{g,\text{best}}$ and $x_{g,\text{worst}}$ denote the current global best and worst individuals found up till the k -th iteration. The diversity threshold is calculated as follows:

$$\zeta = \frac{x_{k,\text{best}} - x_{k,\text{worst}}}{x_{g,\text{best}} - x_{g,\text{worst}}} \quad (31)$$

$$T = e^{\zeta} - 1 \quad (32)$$

Then, the NMA is enabled when $e^{\zeta} - 1 < T_0$, where $T_0 \in [0, e^{-1}]$. The FA would be capable of finding a relatively stable solution close to the target when $e^{\zeta} - 1$ is from 0.0001 to 0.01, that is to say, it is feasible to take T_0 from 10^{-3} to 10^{-5} , and then the step parameter α would be small enough to correspond with the NMA.

4. Numerical Illustrative Examples

In order to demonstrate the superiority of the newly modified algorithm, 18 benchmark functions and a 12-story shear frame model are used to test the performance of the m-NMFA and to provide support for the comparison of the original FA, the GA [39] and the PSO [40] to the modified algorithm.

4.1. Benchmark Work of m-NMFA

Table 1 presents 18 well-known benchmark functions [41,42], and the order number (column 1), the function name (column 2), the formula (column 3), the feasible region (column 4), the minimum value (column 5), and the dimension (column 6) of each function are given successively. These functions can be generally divided into two categories: multimodal functions (f_1 – f_9) and unimodal functions (f_{10} – f_{18}). The multimodal functions are used to prove the ability of the proposed algorithm to escape the local optimal solution and seek the global optimal solution, and the unimodal functions are used to illustrate the precision of the results searched out by the algorithm.

In this test, all the population size and the number of chromosomes are taken to be 30. The maximum number of iterations is taken to be $MaxGen = 1000$. For the FA algorithms, the initial value of the step parameter is taken to be 0.5 in the m-NMFA and 0.25 in the original FA; the attractiveness parameters β_{min} and β_{max} take the values of 0.2 and 1.0, respectively; and the absorption coefficient γ is taken to be 1.0. For the GA, the test uses the built-in GA optimization algorithm of MATLAB R2017a Win64, and most of the input parameters take the default values of the built-in toolbox, except for the number of chromosomes, the maximum number of iterations, and *FunctionTolerance* which is defined as a stop criterion. When the average relative change in the best fitness function value over $MaxGen$ generations is less than or equal to *FunctionTolerance*, the algorithm stops. Given that it demands the same iteration number for all the algorithms, the criterion value should be significantly small and, therefore, takes the value of 10^{-100} in the test. For the PSO, the learning factors take the value of $c_1 = c_2 = 2$, the maximum velocity is taken to be 1.0, and the inertia weighting factor is in a linear form:

$$\omega_t = \omega_2 - t \times \frac{\omega_2 - \omega_1}{MaxGen} \quad (33)$$

where ω_i is the inertia weighting factor of the i -th iteration, and ω_1 and ω_2 are the lower and upper limit values of ω . If the velocity or spatial vector of a particle in one dimension reaches or exceeds the boundary, all dimensions of the particle will be reset to the boundary value. All the functions are run independently using each of the optimization method for 100 times to avoid stochastic discrepancy in MATLAB R2017a Win64, Windows 10 system.

Table 1 shows the results of the m-NMFA under different diversity threshold values, and additionally, the second to fifth columns show the results of the improved algorithm without the NMA. In each case, the mean value, the standard deviation (SD), and the maximum (Max) and minimum (Min) values of the results of each 100 runs are listed. It can be seen from the results that, compared to the case without the NMA, the other three cases are generally better, which proves that the introduction of the NMA can effectively improve the accuracy and stability of the firefly algorithm. In addition, although taking $T_0 = 10^{-1}$ can improve the optimization performance of the algorithm to a certain extent, the results are not good enough. When $T = 10^{-3}$ – 10^{-5} , the accuracy and stability of the optimization algorithm are better than the results of $T = 10^{-1}$, which shows that the start point of the local optimization algorithm in the global optimization algorithm has a significant impact on the optimization result. This test also proves that a diversity threshold value in the range of $T = 10^{-3} \sim 10^{-5}$ is relatively appropriate. In subsequent tests and studies, the m-NMFA will adopt the value of $T = 10^{-5}$.

Table 2 shows the mean values, the standard deviations, and the maximum and minimum values of every 100 runs. It can be seen that the modified algorithm performs generally the same as the other three algorithms in f_1 and f_2 , and it is slightly inferior to the GA in f_7 and f_8 . As for the minimum values of f_{10} and f_{14} , m-NMFA performs slightly worse than PSO and GA. Apart from the cases mentioned above, the m-NMFA significantly outperforms the FA, the GA, and the PSO in both precision and dispersion in most cases. It is worth mentioning that the ability of the m-NMFA to search for the minimum values of multimodal functions is well-illustrated in the tests of f_7 and f_8 for its distinguished minimum values out of the other values.

Table 1. Benchmark functions.

No.	Function Names	Formulations	Limits	Min	D
f_1	Cross-in-Tray Function	$f_1(x) = -0.0001 \left(\left \sin(x_1) \sin(x_2) \exp \left(\left 100 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi} \right \right) \right + 1 \right)^{0.1}$	$x_i \in [-10, 10]$	-2.0626	2
f_2	Schaffer N2 Function	$f_2(x) = 0.5 + \frac{(\sin(x_1^2 - x_2^2))^2 - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	$x_i \in [-100, 100]$	0	2
f_3	Hartmann 6-D Function	$f_3(x) = -\sum_i^4 \alpha_i \exp \left(-\sum_{j=1}^6 A_{ij} (x_j - P_{ij})^2 \right)$, where $\alpha = (1.0, 1.2, 3.0, 3.2)^T$ $A = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix}$ $P = 10^{-4} \begin{pmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{pmatrix}$	$x_i \in (0, 1)$	-3.3224	6
f_4	Zakharov Function	$f_4(x) = \sum_{i=1}^D x_i^2 + \left(\sum_{i=1}^D 0.5ix_i \right)^2 + \left(\sum_{i=1}^D 0.5ix_i \right)^4$	$x_i \in [-5, 10]$	0	30
f_5	Alpine Function	$f_5(x) = \sum_{i=1}^D x_i \sin(x_i) + 0.1x_i $	$x_i \in [-10, 10]$	0	30
f_6	Griewank Function	$f_6(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$x_i \in [-600, 600]$	0	30
f_7	Penalized 1 Function	$f_7(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 \left[1 + 10 \sin^2(\pi y_{i+1}) \right] + (y_D - 1)^2 \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$, where $y_i = 1 + \frac{1}{4}(x_i + 1)$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$x_i \in [-50, 50]$	0	30
f_8	Penalized 2 Function	$f_8(x) = \frac{1}{10} \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 \left[1 + \sin^2(3\pi x_{i+1}) \right] + (x_D - 1)^2 \left[1 + \sin^2(2\pi x_D) \right] \right\} + \sum_{i=1}^D u(x_i, 5, 100, 4)$,	$x_i \in [-50, 50]$	0	30

Table 1. Cont.

No.	Function Names	Formulations	Limits	Min	D
f_9	Ackley Function	$f_9(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)}\right) + 20 + e$	$x_i \in [-32.768, 32.768]$	0	30
f_{10}	Sum of Different Powers Function	$f_{10}(x) = \sum_{i=1}^D x_i ^{i+1}$	$x_i \in [-1, 1]$	0	30
f_{11}	Sphere Function	$f_{11}(x) = \sum_{i=1}^D x_i^2$	$x_i \in [-5.12, 5.12]$	0	30
f_{12}	Sum Squares Function	$f_{12}(x) = \sum_{i=1}^D i x_i^2$	$x_i \in [-5.12, 5.12]$	0	30
f_{13}	Rosenbrock Function	$f_{13}(x) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$x_i \in [-2.048, 2.048]$	0	30
f_{14}	Dixon-Price Function	$f_{14}(x) = (x_1 - 1)^2 + \sum_{i=2}^D i(2x_i^2 - x_{i-1})^2$	$x_i \in [-10, 10]$	0	30
f_{15}	Rotated Hyper-Ellipsoid Function	$f_{15}(x) = \sum_{i=1}^D \sum_{j=1}^i x_j^2$	$x_i \in [-65.536, 65.536]$	0	30
f_{16}	Perm Function 0, d, β	$f_{16}(x) = \sum_{i=1}^D \left(\sum_{j=1}^D (j+10) \left(x_j^i - \frac{1}{j^i} \right) \right)^2$	$x_i \in [-30, 30]$	0	30
f_{17}	Schwefel 1.2 Function	$f_{17}(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	$x_i \in [-100, 100]$	0	30
f_{18}	Schwef 2.22 Function	$f_{18}(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$x_i \in [-100, 100]$	0	30

Figure 2 shows the convergence diagrams of the 18 functions operated by the FA, the GA, the PSO, and the m-NMFA. The value of the ordinate is the mean value after 100 operations. It can be seen that the m-NMFA converges faster in most cases and has a strong high-precision solution searching ability.

4.2. Numerical Simulation of Shear Frame

A 12-story steel shear frame shown in Figure 3 is used to test the performance of the m-NMFA. The frame is assumed to be fixed on a solid ground and is of 1.8 m in height. The masses are uniformly distributed on each floor and are totally supported by six full-length rectangular columns with an identical cross-section of 6 mm in thickness and 25 mm in width. The mass of each floor is taken to be $m_e = 75 \text{ kg}$, and the inter-story stiffness is calculated to be $k_e = 1.512 \times 10^5 \text{ N/m}$. Hence, the nominal undamaged structure mass and stiffness matrices are calculated as follows:

$$M_u = \begin{bmatrix} m_e & & & & & \\ & m_e & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & m_e & \\ & & & & & m_e \end{bmatrix}_{12 \times 12} \tag{34}$$

$$K_u = \begin{bmatrix} 2k_e & -k_e & & & & \\ -k_e & 2k_e & -k_e & & & \\ & -k_e & \ddots & \ddots & & \\ & & \ddots & \ddots & -k_e & \\ & & & \ddots & -k_e & 2k_e \end{bmatrix}_{12 \times 12} \tag{35}$$

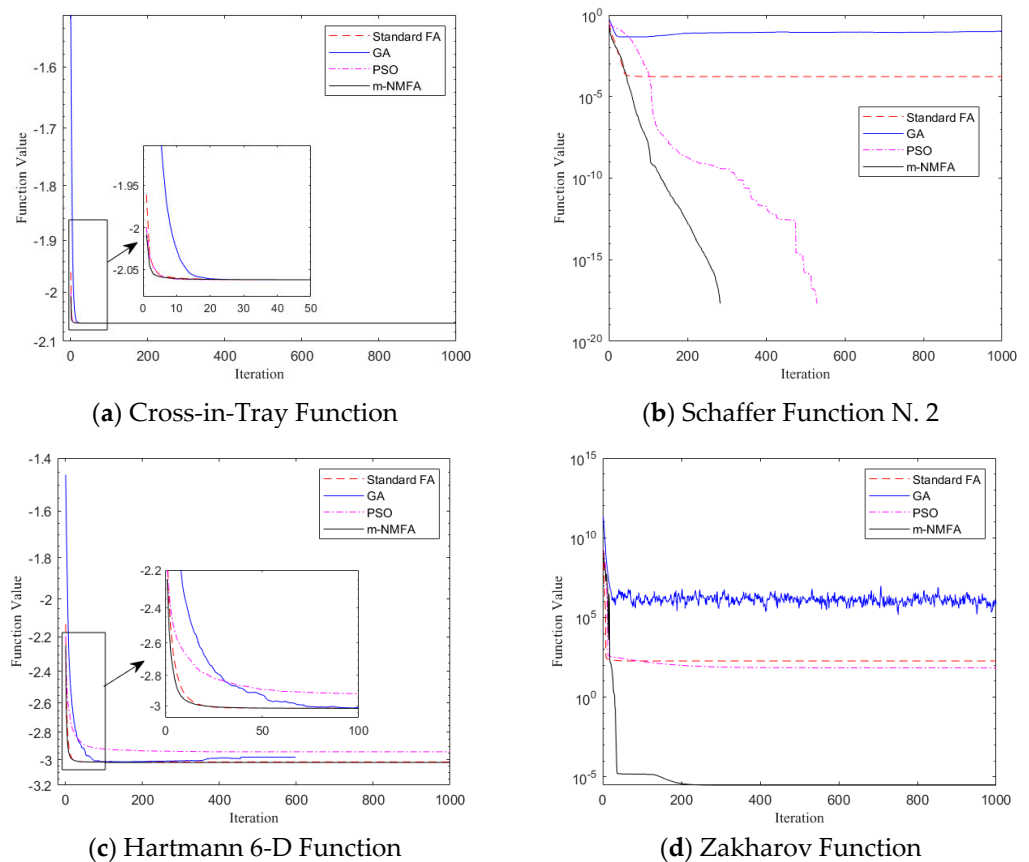
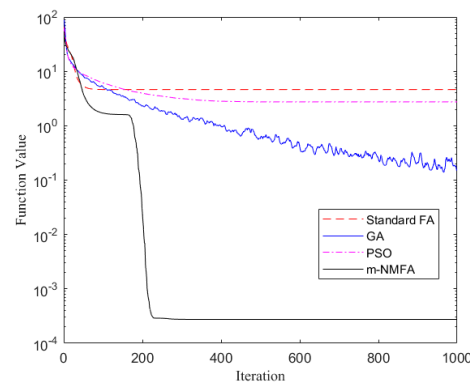
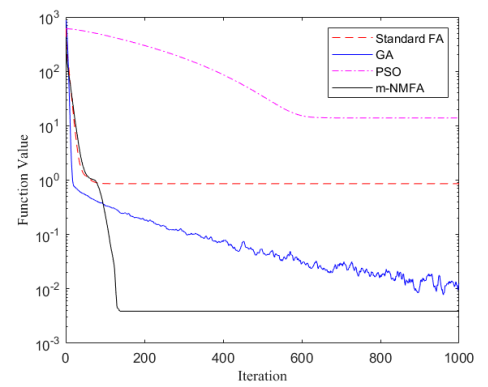


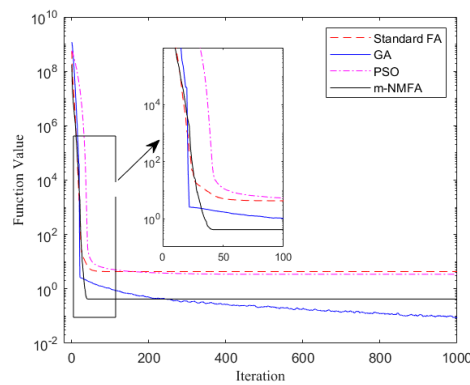
Figure 2. Cont.



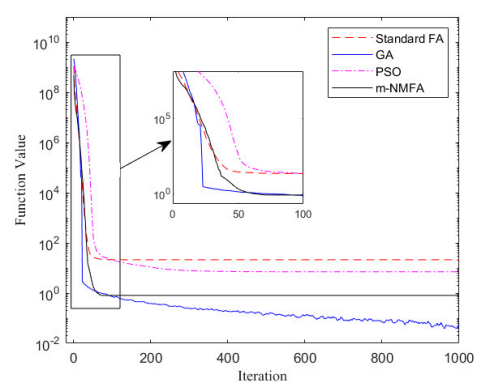
(e) Alpine 1 Function



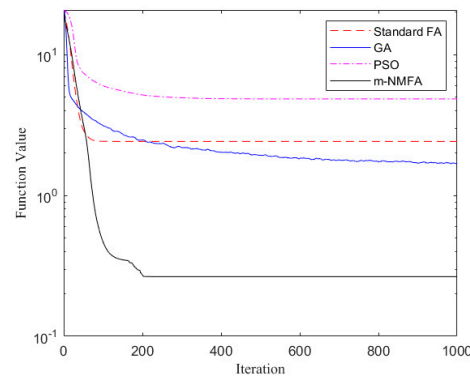
(f) Griewank Function



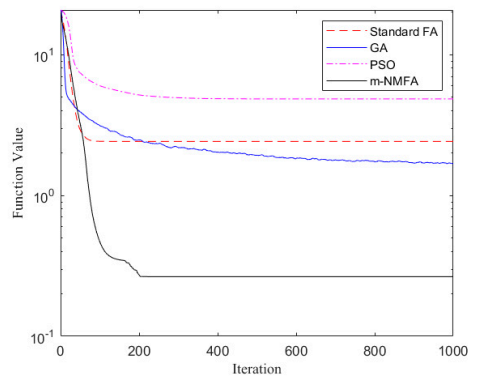
(g) Penalized 1 Function



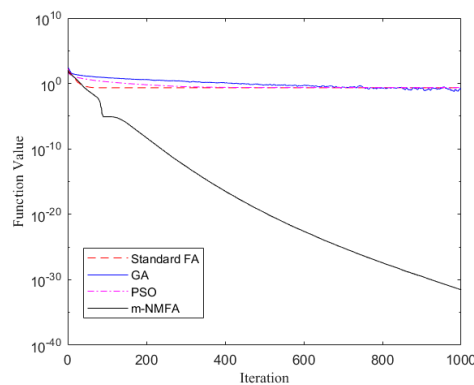
(h) Penalized 2 Function



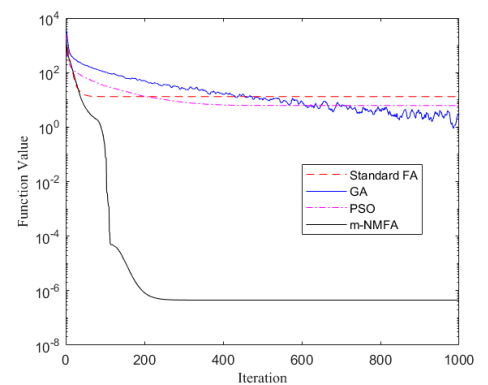
(i) Ackley Function



(j) Sum of Different Powers Function



(k) Sphere Function



(l) Sum Squares Function

Figure 2. Cont.

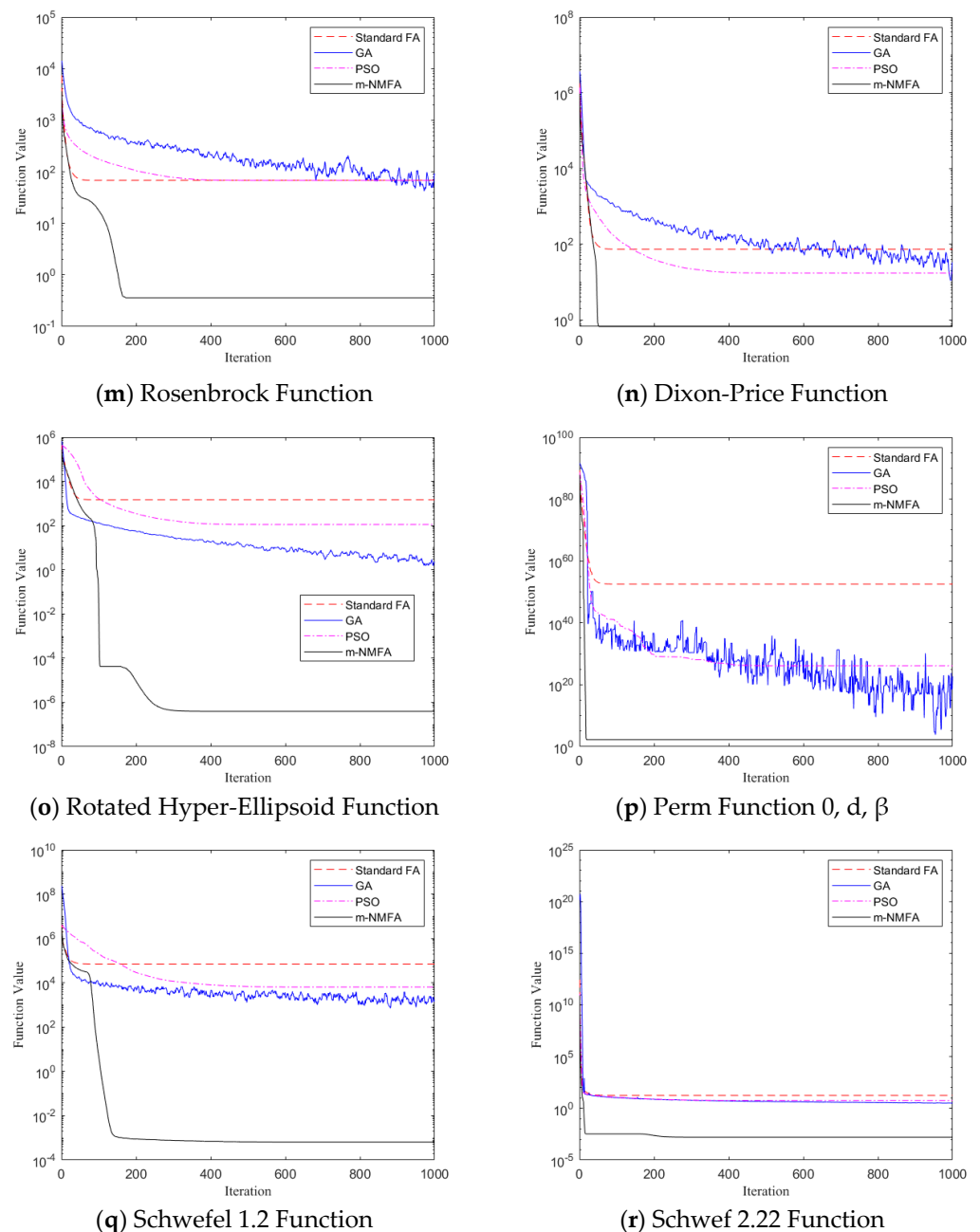


Figure 2. Convergence diagram of four optimization algorithms.

It is assumed to have a stiffness reduction of overall 20%, 40%, and 20% on the 5th, 6th, and 7th inter-story columns, which is denoted by the artificially settled damage vector θ_{ith} ($i = 5, 6, 7$). Gaussian white noises with mean values of zero are considered for the modal data, and the degrees of which are taken to be 1% for modal frequencies and 3% for modal shapes. Complete and incomplete measurements are considered. In the complete measurement case, all the first 12 modal parameters can be well measured. In the incomplete case, it is assumed that only the first eight modal parameters and the special displacement information of the 1st, 2nd, 4th, 6th, 8th, 10th, 11th, and 12th stories can be obtained. In order to eliminate accidental errors, the tests are executed independently for $N_t = 100$ times and 100 sets of modal parameters are used. The simulation progress is operated using the FA, the GA, the PSO, and the m-NMFA separately and the algorithm parameters are kept the same as those in Section 4.1.

In this example, the standard deviation (SD) and the coefficient of variances (CV) are calculated for each parameter, which are listed in Tables 2 and 3. Figure 4 shows the results of the damage vector θ for both cases. It can be seen that, in the complete measurement case, all the algorithms are able to identify the position and the degree of the damage ideally. However, in the incomplete measurement case, only the m-NMFA can achieve the goal of identifying the damage of a structure.

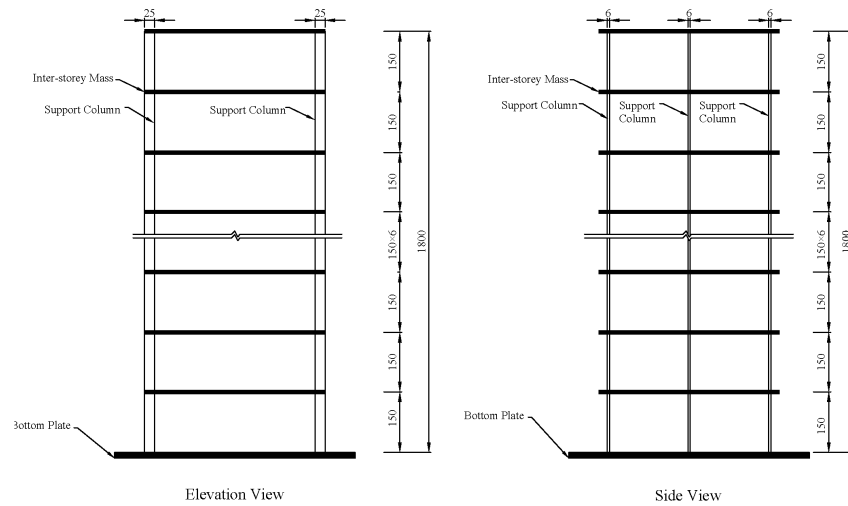


Figure 3. Schematic diagram of a 12-story steel shear frame (Unit: mm).

Table 2. The results of shear frame: complete measurement case.

No.	θ_a	θ	FA		θ	GA		θ	PSO		m-NMFA		
			SD	CV		SD	CV		SD	CV	θ	SD	CV
1	0	0.026	0.0012	0.0452	0.0083	0.0011	0.1374	-0.0030	0.0020	0.6884	0.0081	0.0011	0.1408
2	0	0.007	0.0016	0.2196	-0.0053	0.0015	0.2858	-0.0104	0.0029	0.2772	-0.0054	0.0015	0.2785
3	0	0.032	0.0014	0.0452	-0.0044	0.0013	0.2932	0.0004	0.0042	9.3279	-0.0047	0.0013	0.2729
4	0	0.042	0.0015	0.0350	0.0061	0.0013	0.2196	0.0414	0.0043	0.1042	0.0058	0.0013	0.2291
5	-0.2	-0.185	0.0012	0.0065	-0.1932	0.0012	0.0060	-0.2016	0.0024	0.0121	-0.1933	0.0012	0.0060
6	-0.4	-0.404	0.0008	0.0019	-0.4036	0.0008	0.0019	-0.4124	0.0015	0.0036	-0.4036	0.0008	0.0019
7	-0.2	-0.205	0.0012	0.0058	-0.1992	0.0012	0.0061	-0.2025	0.0026	0.0127	-0.1993	0.0012	0.0061
8	0	0.028	0.0015	0.0553	-0.0058	0.0014	0.2402	-0.0063	0.0027	0.4338	-0.0061	0.0014	0.2299
9	0	0.032	0.0013	0.0416	-0.0030	0.0012	0.3932	-0.0235	0.0030	0.1271	-0.0031	0.0012	0.3779
10	0	0.046	0.0021	0.0452	-0.0060	0.0017	0.2859	-0.0019	0.0030	1.5957	-0.0062	0.0017	0.2760
11	0	0.015	0.0015	0.0943	-0.0079	0.0014	0.1722	0.0127	0.0040	0.3176	-0.0080	0.0014	0.1688
12	0	-0.001	0.0024	2.7776	-0.0121	0.0022	0.1850	0.0621	0.0065	0.1043	-0.0120	0.0022	0.1868

Table 3. The results of shear frame: incomplete measurement case.

No.	θ_a	θ	FA		θ	GA		θ	PSO		m-NMFA		
			SD	CV		SD	CV		SD	CV	θ	SD	CV
1	0	0.017	0.0024	0.0293	0.0045	0.0020	0.0459	0.0613	/	1.2185	-0.0026	0.0020	0.7461
2	0	0.0473	0.0015	0.0368	0.0383	0.0077	0.0150	0.1942	/	0.6339	0.0070	0.0044	0.6253
3	0	0.0436	/	0.0039	0.0036	0.0023	0.2470	0.1399	0.0023	0.0077	0.0199	0.0081	0.4085
4	0	0.0117	/	0.0244	0.0271	/	0.0141	0.1616	0.0097	0.0346	-0.0272	0.0062	0.2278
5	-0.2	-0.0973	0.0121	0.3250	-0.1704	0.0015	0.0026	0.1186	/	0.4796	-0.2036	0.0037	0.0184
6	-0.4	-0.3829	0.0038	0.0348	-0.3727	0.0069	0.0104	-0.3700	/	0.0330	-0.4072	0.0037	0.0091
7	-0.2	-0.2059	0.0037	0.0098	-0.0327	0.0018	0.0041	-0.0968	/	0.0248	-0.1977	0.0047	0.0235
8	0	0.0682	0.0036	0.1014	0.0266	0.0038	0.0056	0.1807	/	0.0366	-0.0156	0.0089	0.5712
9	0	-0.0001	0.0022	0.0083	0.0813	0.0048	0.0841	0.1135	/	0.4384	-0.0244	0.0071	0.2907
10	0	0.0468	0.0067	0.0263	-0.0314	0.0037	0.2940	0.0855	/	0.2556	0.0058	0.0062	1.0808
11	0	0.0299	0.0043	0.0431	0.0151	0.0045	0.0155	0.1274	0.0018	0.0393	0.0046	0.0027	0.5872
12	0	0.0211	0.0030	0.0867	0.0336	0.0018	0.0134	0.1086	/	0.0620	-0.0079	0.0030	0.3828

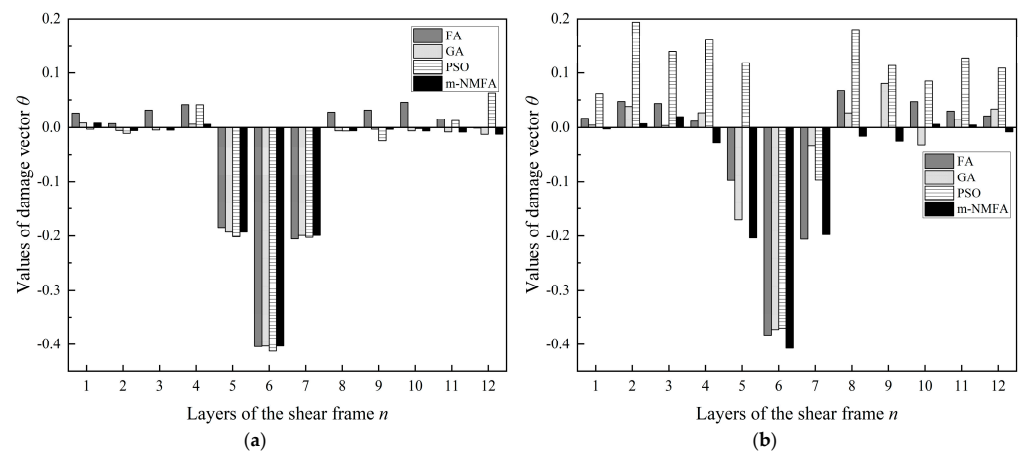


Figure 4. The results of damage vector θ . (a) Complete measurement case. (b) Incomplete measurement case.

5. Experimental Illustrative Example

A six-story steel shear frame is designed to perform a vibration experiment, in which the modal parameters are identified by FDD to form a modal flexibility matrix. Then, the m-NMFA is utilized to search out the minimum value of the modal-flexibility-based objective function. Damage identification problem is introduced to illustrate the effectiveness of the proposed method.

The structure is fixed on the ground by foundation bolts. The length, width, and thickness of the interlayer plates are 450 mm, 450 mm, and 10 mm, respectively. The thickness center spacing of each plate is 250 mm. The interlayer stiffness is provided by the main columns fixed at the four vertices of the plates and additional columns at the midpoint of the side edges. An angle steel is welded with a steel plate in the center of it. An external excitation of random force and random position is applied to the structure by hammering excitation. The displacement response of each layer is measured using an IL-300 laser displacement meter. The measuring point is set at the center of the angle steel baffle. The sampling frequency of the meter is 200 Hz. The masses from m1 to m6 are 17.151 kg, 17.123 kg, 17.215 kg, 17.169 kg, 17.193 kg, and 16.731 kg, respectively, and the nominal interlaminar stiffness is 81920n/m. Before the test, a lumped mass reference finite element model is established according to the measured mass of each layer and the calculated nominal stiffness.

The displacement responses of the steel shear frames in the damaged and undamaged states are tested. In the damaged state, the additional columns of the fifth floor are missing, of which the nominal damage coefficient is -20% . In both tests, the responses of the 1st, 2nd, 4th, 5th, and 6th stories are measured, and the measurement repeats 50 times for each test and lasts for 5 min each. Figures 5 and 6 show the time domain information of the undamaged and damaged structures.

The frequencies of the first six modes of the frame structure are shown in Table 4, which take the average values of 50 measurements. The mode shapes shown in Figure 7 are normalized with respect to the reduced mass matrix M^m .

Table 4. The first six frequencies of the shear frame identified by FDD (Unit: Hz).

Modal Order	Undamaged	Damaged
1	2.6616	2.6104
2	7.8329	7.3901
3	12.5790	12.3932
4	16.6569	16.5127
5	19.7584	19.0000
6	21.4556	21.2004

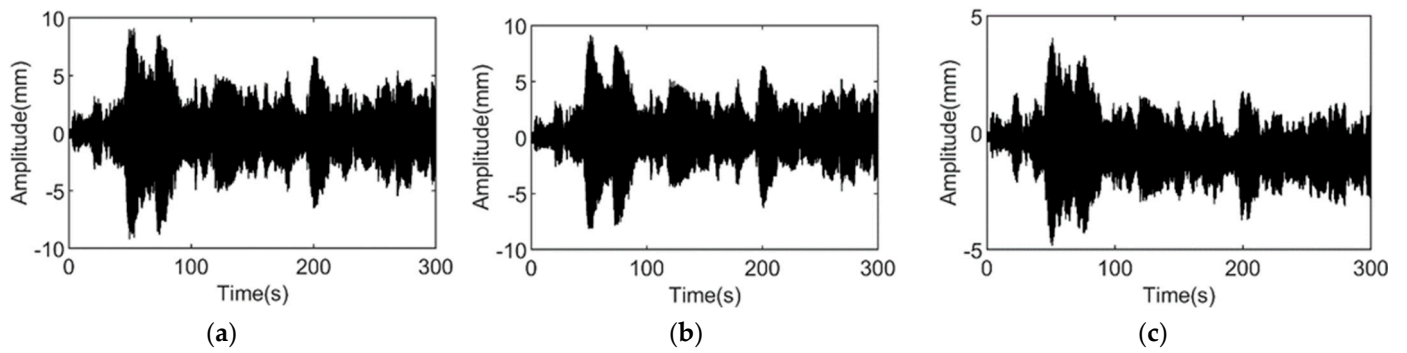


Figure 5. Time domain information of undamaged structure vibration. (a) The 1st story; (b) the 2nd story; and (c) the 4th story.

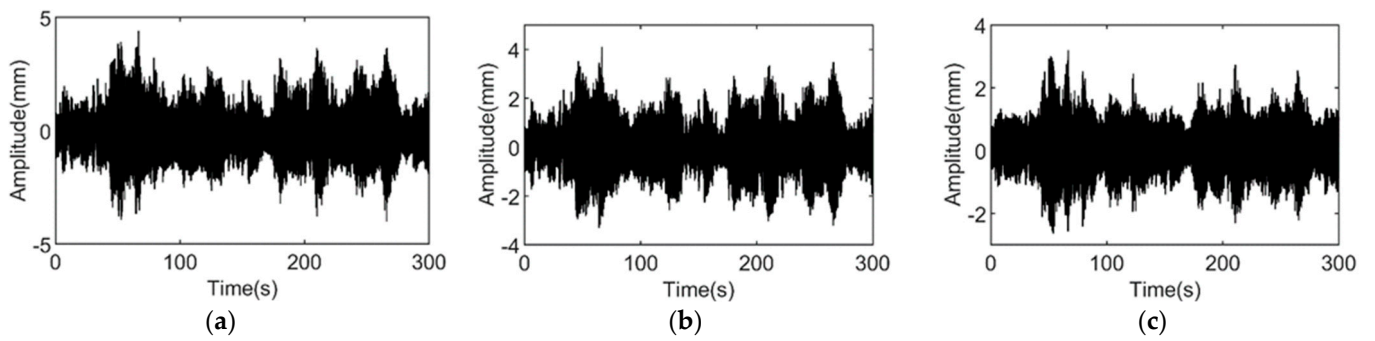


Figure 6. Time domain information of damaged structure vibration. (a) The 1st story; (b) the 2nd story; and (c) the 4th story.

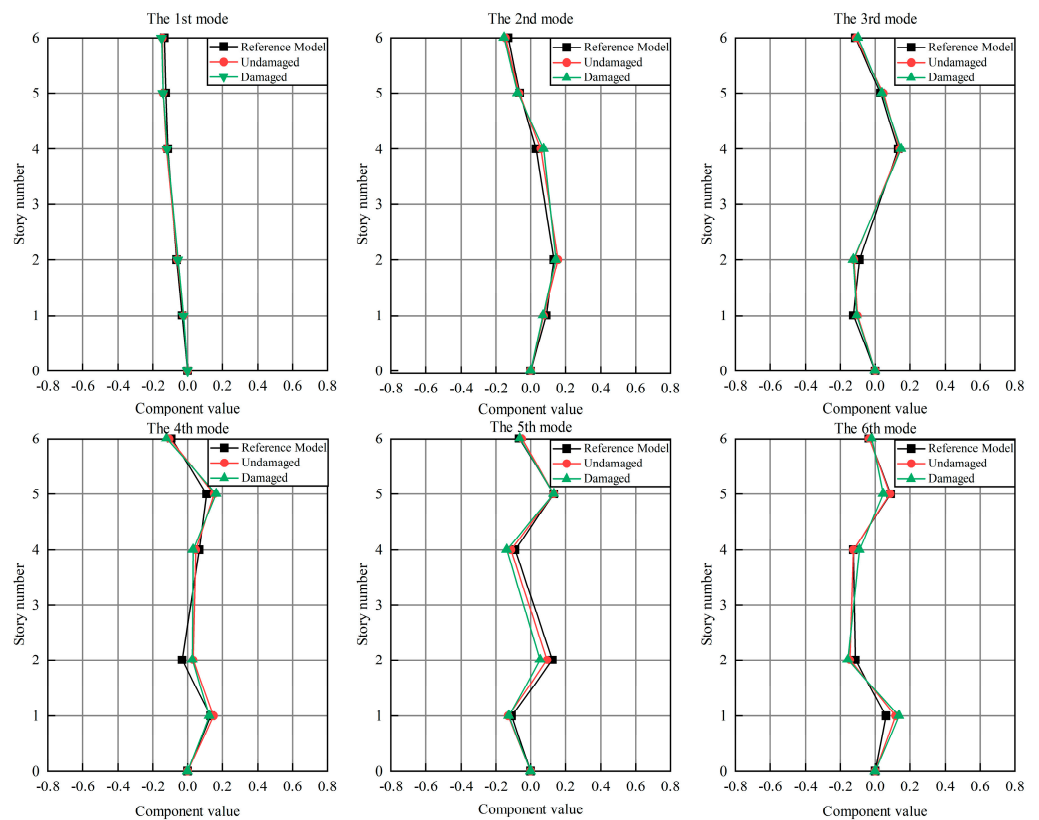


Figure 7. The first six mode shapes of steel shear frames identified by FDD.

In the Bayesian probabilistic approach, the probability distributions of the model updating parameters of the damaged and undamaged structures are needed to calculate the cumulative probability distribution of the damage parameters in the damaged state. The distributions of the damage parameters can be obtained using the following formula:

$$P_n(d) = P\left((1 + \theta_n^d) < (1 - d)(1 + \theta_n^{ud})\right) \approx \Phi\left(\frac{(1 - d)(1 + \hat{\theta}_n^{ud}) - (1 + \hat{\theta}_n^d)}{\sqrt{(1 - d)^2(\sigma_n^{ud})^2 + (\sigma_n^d)^2}}\right) \quad (36)$$

where the damage extent d is used to describe the value of the stiffness reduction parameter; θ_n^d and θ_n^{ud} are the model updating parameters of the damaged and undamaged structures; and $\hat{\theta}_n^d$, $\hat{\theta}_n^{ud}$, $\hat{\sigma}_n^d$ and $\hat{\sigma}_n^{ud}$ are the mean values and standard deviations of the model updating parameters in Gaussian probability distribution. The value of θ_n can be obtained by choosing the mean value of each distribution in the probability density function diagram. The standard deviations of the n -th damage parameters can be calculated using the following formula:

$$\sigma_n = \frac{\sum_i^\infty (d_{n,i} - \theta_n)^2 y(x_{n,i}) \Delta d}{\sum_j^\infty y(d_{n,j}) \Delta d} \quad (37)$$

where $\Delta d = d_{n,v+1} - d_{n,v}$, $v \in \{i, j | i \in \mathbb{N}^*, j \in \mathbb{N}^*\}$, and $y(d)$ is the Gaussian probability density function of d .

The model updating parameters of the finite element model of the undamaged structure are identified using the m-NMFA, which are shown in Table 5. The model updating parameters of the damaged structure are shown in Table 6.

Table 5. Model updating parameters of the undamaged structure using the Bayesian probability method.

n	1	2	3	4	5	6
$\hat{\theta}_n^{ud}$	0.4402	−0.0904	−0.1299	−0.1259	−0.0910	−0.0126
$\hat{\sigma}_n^{ud}$	0.0051	0.0022	0.0033	0.0030	0.0028	0.0023

Table 6. Model updating parameters of the damaged structure using the Bayesian probability method.

n	1	2	3	4	5	6
$\hat{\theta}_n^d$	0.4923	−0.0792	−0.1438	−0.2008	−0.3099	−0.0250
$\hat{\sigma}_n^d$	0.0050	0.0022	0.0034	0.0032	0.0018	0.0023

According to the identification results shown in Tables 5 and 6, the cumulative probability distribution of the structural damage parameter of each floor is calculated using Formula 35. The cumulative probability distribution is plotted in Figure 8. The probability density function is plotted in Figure 9. Figure 10 shows the mean value of each distribution in the probability density function directly.

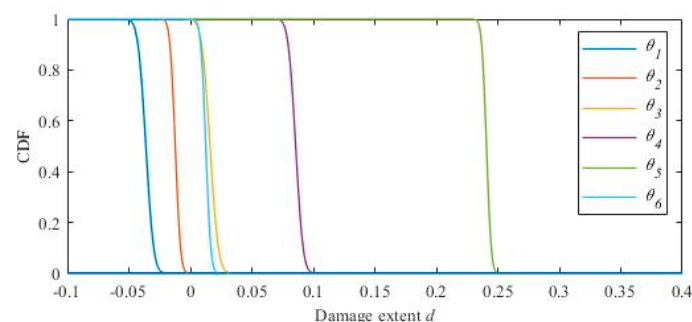


Figure 8. The cumulative probability function of the damage parameters.

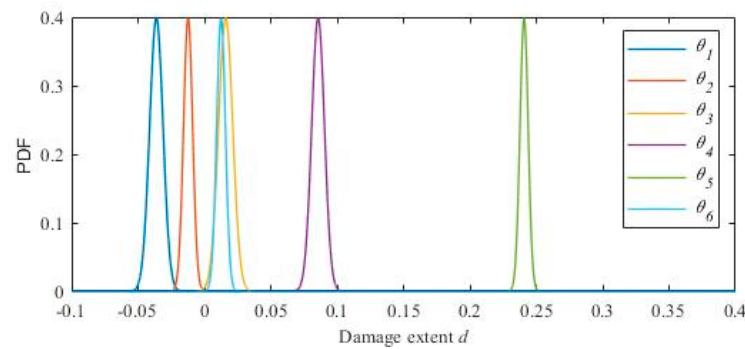


Figure 9. The probability density function of the damage parameters.

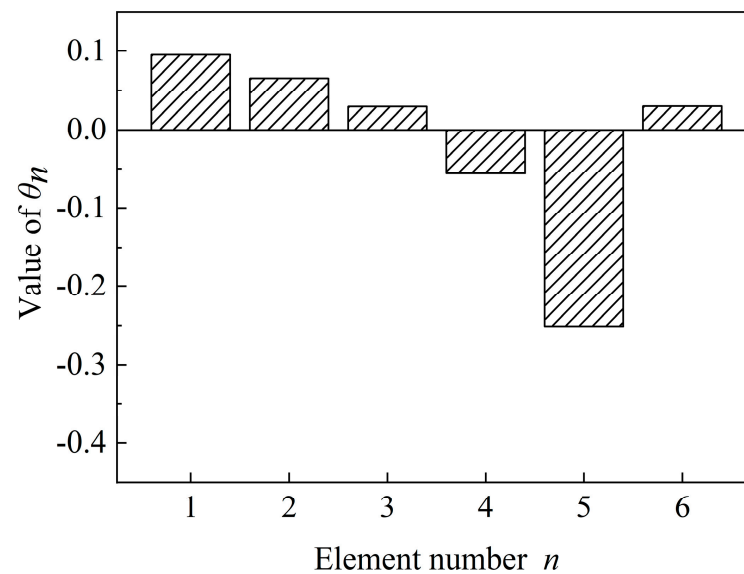


Figure 10. The damage parameter of the shear frame.

6. Conclusions

Based on the modal flexibility matrix, this paper uses the Bayesian probability method to establish the objective function for the model updating problem, which is solved using the proposed m-NMFA. The results show that this method can accurately fix the damage position and determine the damage extent.

Compared to the other meta-heuristic optimization algorithms, the proposed m-NMFA has a stronger optimization ability in solving benchmark functions and the finite element model updating objective functions, which shows a higher applicability in the practical problems of model updating. The reason why the m-NMFA has more superior optimization performance is mainly due to the following three points:

1. In the research on the random step size formula, it is found that the smaller the random step size is at the beginning of the iteration, the faster the convergence speed of the algorithm will be. However, as the random step size gradually approaches zero at the end of the iteration, its value should be large enough to keep the algorithm with sufficient exploration ability.
2. In the research on the value of diversity threshold, a fraction is used to quantify the diversity of the population to enable the NMA algorithm to start at an appropriate time. The diversity threshold is quantized into a value between $(0, e^{-1}]$ through exponential form. When the diversity threshold is taken to be between $10^{-3} \sim 10^{-5}$, the algorithm can obtain a more accurate optimal solution.
3. The selection of the iterative formula of the attraction parameter has great impact on the solving ability of the FA in solving the multi-dimensional optimization problems,

The selection of the iterative formula of the attraction parameter has a great impact on the ability of the FA in solving multi-dimensional optimization problems, which would lead to stagnation and non-convergence if an improper formula were selected. This paper avoids such problem by selecting an appropriate formula for the coefficient of attraction.

In the research of model updating based on modal flexibility and Bayesian method, the modal flexibility theory, the system equivalent reduction and expansion process, and the basic theory of Bayesian method are introduced. The objective function established based on these theories provides a relatively complete theoretical support for solving the model updating problems of limited measuring points and number of modes, so that people can obtain reliable results from limited structural information. Bayesian method allows people to empirically select the prior probability distribution of the modified parameters so as to exert influence on the identification results, which provides theoretical and practical convenience for solving the problems of model updating. Through the numerical simulation of a 12-story shear frame and the experimental test of a 6-story steel shear frame, the reliability of the Bayesian probability method in solving model updating and structural damage identification is verified.

Author Contributions: Conceptualization, Z.F.; methodology, Z.F. and W.W.; software, W.W.; validation, W.W.; formal analysis, J.Z.; investigation, W.W.; resources, Z.F.; data curation, W.W.; writing—original draft preparation, W.W. and J.Z.; writing—review and editing, Z.F. and J.Z.; visualization, W.W.; supervision, Z.F.; project administration, Z.F.; funding acquisition, Z.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of China (grant No. 52178284, 51708203), the Natural Science Foundation of Chongqing (grant No. 2022NSCQ-MSX5727), and the Natural Science Foundation of Hunan Province (grant No. 2021JJ30106).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data analyzed in this study have been included in this paper.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table 1. The results of the improved algorithm under different diversity threshold values.

No.	Modified Algorithm with No NMA				$T = 10^{-1}$				$T = 10^{-3}$				$T = 10^{-5}$			
	Mean	SD	Max	Min	Mean	SD	Max	Min	Mean	SD	Max	Min	Mean	SD	Max	Min
f_1	-2.063	2.68×10^{-15}	-2.063	-2.063	-2.063	2.68×10^{-15}	-2.063	-2.063	-2.063	2.68×10^{-15}	-2.063	-2.063	-2.063	2.68×10^{-15}	-2.063	-2.063
f_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_3	-3.022	0.032	-2.942	-3.042	-3.025	0.028	-2.981	-3.042	-3.012	0.031	-2.981	-3.042	-3.021	0.029	-2.981	-3.042
f_4	73.192	28.043	145.588	19.536	2.90×10^{-6}	2.12×10^{-6}	1.30×10^{-5}	4.38×10^{-7}	2.61×10^{-6}	2.25×10^{-6}	1.48×10^{-5}	3.05×10^{-7}	2.71×10^{-6}	1.99×10^{-6}	8.48×10^{-6}	1.95×10^{-7}
f_5	1.703	1.369	7.756	0.033	2.39×10^{-4}	1.49×10^{-4}	7.06×10^{-4}	1.67×10^{-5}	2.13×10^{-4}	1.39×10^{-4}	6.57×10^{-4}	7.92×10^{-6}	2.71×10^{-4}	1.74×10^{-4}	7.94×10^{-4}	2.14×10^{-5}
f_6	4.91×10^{-3}	6.50×10^{-3}	0.028	3.77×10^{-6}	0.032	0.064	0.460	1.16×10^{-13}	0.035	0.034	0.162	6.91×10^{-13}	3.89×10^{-3}	7.00×10^{-3}	0.044	1.32×10^{-13}
f_7	0.957	1.102	7.171	4.62×10^{-4}	3.727	5.407	26.387	8.15×10^{-15}	1.650	3.082	22.824	9.36×10^{-15}	0.423	0.621	3.720	1.05×10^{-15}
f_8	0.155	1.251	12.226	2.03×10^{-5}	26.232	35.036	100.531	5.94×10^{-16}	2.743	9.077	55.762	6.46×10^{-14}	0.808	4.779	45.357	9.50×10^{-15}
f_9	0.262	0.448	1.900	7.99×10^{-15}	3.955	1.580	17.586	2.317	0.240	0.483	1.900	4.44×10^{-15}	0.265	0.502	1.778	4.44×10^{-15}
f_{10}	2.90×10^{-6}	2.60×10^{-6}	1.69×10^{-5}	2.55×10^{-7}	8.93×10^{-78}	3.94×10^{-77}	3.64×10^{-76}	1.74×10^{-84}	2.20×10^{-71}	2.20×10^{-70}	2.20×10^{-69}	7.04×10^{-84}	3.68×10^{-72}	3.68×10^{-71}	3.68×10^{-70}	5.99×10^{-83}
f_{11}	2.96×10^{-8}	1.64×10^{-7}	1.17×10^{-6}	1.39×10^{-32}	7.52×10^{-15}	7.52×10^{-14}	7.52×10^{-13}	1.72×10^{-32}	2.89×10^{-32}	7.39×10^{-33}	4.76×10^{-32}	1.65×10^{-32}	1.84×10^{-15}	1.84×10^{-14}	1.84×10^{-13}	1.56×10^{-32}
f_{12}	1.558	1.552	6.694	0.044	3.62×10^{-7}	4.53×10^{-7}	2.13×10^{-6}	1.85×10^{-9}	3.80×10^{-7}	4.71×10^{-7}	2.56×10^{-6}	1.19×10^{-9}	3.36×10^{-7}	4.60×10^{-7}	2.73×10^{-6}	9.01×10^{-10}
f_{13}	29.276	7.036	77.985	22.668	0.997	1.735	3.987	4.51×10^{-13}	0.797	1.603	3.987	1.81×10^{-12}	0.678	1.505	3.987	5.81×10^{-12}
f_{14}	8.061	9.410	65.806	0.681	0.667	2.04×10^{-7}	0.667	0.667	0.667	0	0.667	0.667	0.667	6.21×10^{-8}	0.667	0.667
f_{15}	118.714	131.326	674.149	0.563	3.09×10^{-7}	3.91×10^{-7}	1.82×10^{-6}	5.69×10^{-10}	3.12×10^{-7}	4.06×10^{-7}	2.05×10^{-6}	1.35×10^{-10}	2.95×10^{-7}	4.39×10^{-7}	2.48×10^{-6}	2.02×10^{-10}
f_{16}	1.74×10^{26}	1.04×10^{27}	8.19×10^{27}	8.19×10^{27}	209.091	326.701	1.42×10^3	2.53×10^{-8}	135.316	229.458	1.50×10^3	2.71×10^{-7}	187.578	296.558	1.42×10^3	9.43×10^{-8}
f_{17}	2.41×10^4	1.58×10^4	8.91×10^4	4.35×10^3	6.13×10^{-4}	4.51×10^{-4}	2.48×10^{-3}	7.10×10^{-5}	7.22×10^{-4}	6.88×10^{-4}	5.09×10^{-3}	1.08×10^{-4}	6.11×10^{-4}	3.91×10^{-4}	1.99×10^{-3}	1.32×10^{-4}
f_{18}	2.444	2.584	16.389	0.041	1.68×10^{-3}	1.18×10^{-3}	5.81×10^{-3}	7.15×10^{-6}	1.70×10^{-3}	1.16×10^{-3}	5.61×10^{-3}	7.39×10^{-5}	1.73×10^{-3}	1.09×10^{-3}	6.50×10^{-3}	2.66×10^{-5}

Table 2. Comparison of the four optimization algorithms.

No.	FA				GA				PSO				m-NMFA			
	Mean	SD	Max	Min	Mean	SD	Max	Min	Mean	SD	Max	Min	Mean	SD	Max	Min
f_1	-2.0626	2.65×10^{-15}	-2.0626	-2.0626	-2.0626	1.01×10^{-10}	-2.0626	-2.0626	-2.0626	2.68×10^{-15}	-2.0626	-2.0626	-2.0626	2.68×10^{-15}	-2.0626	-2.0626
f_2	3.53×10^{-5}	3.26×10^{-4}	3.25×10^{-3}	0.00×10^0	0.046	0.0982	0.475	2.95×10^{-14}	0	0	0	0	0	0	0	0
f_3	-3.0197	0.0366	-2.9233	-3.0425	-3.0179	0.0303	-2.9799	-3.0425	-2.9622	0.0871	-2.6175	-3.0425	-3.0210	0.0295	-2.9810	-3.0425
f_4	193.2893	48.4340	328.8250	91.7273	61.9113	71.3777	318.4053	1.5917	76.3396	77.5750	528.4291	1.1334	2.71×10^{-6}	1.99×10^{-6}	8.48×10^{-6}	1.95×10^{-7}
f_5	4.7192	2.4243	12.9291	0.9156	0.1317	0.2206	1.7274	3.80×10^{-3}	3.2038	1.9981	11.9646	0.5124	2.71×10^{-4}	1.74×10^{-4}	7.94×10^{-4}	2.14×10^{-5}
f_6	0.8609	0.4527	2.8608	0.0800	4.13×10^{-3}	9.34×10^{-3}	0.0620	4.08×10^{-5}	13.8875	4.8491	30.1011	4.08×10^{-5}	3.89×10^{-3}	7.00×10^{-3}	0.0443	1.32×10^{-13}
f_7	4.3163	2.2529	12.2211	0.4024	0.0667	0.0862	0.3263	4.56×10^{-5}	3.4507	1.1366	7.0231	4.56×10^{-5}	0.4225	0.6215	3.7198	1.05×10^{-15}
f_8	21.3056	16.1388	122.1799	1.0126	0.0259	0.0344	0.2279	2.73×10^{-4}	7.0522	10.4476	49.7862	2.73×10^{-4}	0.8075	4.7795	45.3568	9.50×10^{-15}
f_9	2.4178	0.9188	4.9678	0.6201	1.6640	0.5075	2.9591	9.45×10^{-2}	4.8442	1.2063	7.6330	0.0945	0.2651	0.5021	1.7780	4.44×10^{-15}
f_{10}	1.94×10^{-5}	3.23×10^{-5}	2.21×10^{-4}	1.14×10^{-6}	6.81×10^{-4}	2.05×10^{-3}	0.0159	7.55×10^{-8}	4.98×10^{-6}	1.51×10^{-5}	1.08×10^{-4}	0	3.68×10^{-72}	3.68×10^{-71}	3.68×10^{-70}	5.99×10^{-83}
f_{11}	0.2330	0.1597	0.8486	0.0507	0.0193	0.0219	0.1432	8.46×10^{-4}	0.2220	0.1488	0.8418	0.0385	1.84×10^{-15}	1.84×10^{-14}	1.84×10^{-13}	1.56×10^{-32}
f_{12}	15.7275	10.1891	50.2489	1.5753	0.5308	0.6093	3.2460	0.0150	7.0292	4.2909	25.7904	0.3848	3.36×10^{-7}	4.60×10^{-7}	2.73×10^{-6}	9.01×10^{-10}
f_{13}	73.6154	26.5540	148.5464	31.3071	38.1670	40.1835	135.4710	0.2690	70.8902	50.9525	496.6682	30.3306	0.6777	1.5051	3.9866	5.81×10^{-12}
f_{14}	86.3426	153.5819	1.46×10^3	6.6724	6.7756	5.5898	33.0558	0.3401	18.2969	19.1600	125.8380	2.8606	0.6667	6.21×10^{-8}	0.6667	0.6667
f_{15}	1.60×10^3	1.35×10^3	7.49×10^3	80.0058	0.6165	0.8272	4.9430	0.0178	106.0857	99.0572	678.9951	9.3399	2.95×10^{-7}	4.39×10^{-7}	2.48×10^{-6}	2.02×10^{-10}
f_{16}	2.04×10^{31}	1.96×10^{32}	1.96×10^{33}	3.12×10^{12}	8.54×10^{11}	7.48×10^{12}	7.40×10^{13}	1.69×10^{-3}	3.37×10^{25}	2.51×10^{26}	2.38×10^{27}	1.92×10^3	187.5780	296.5578	1.42×10^3	9.43×10^{-8}
f_{17}	7.361×10^4	5.340×10^4	2.837×10^5	1.429×10^4	180.9207	674.6373	6.31×10^3	9.3670	6.02×10^3	3.85×10^3	2.44×10^4	961.8477	6.11×10^{-4}	3.91×10^{-4}	1.99×10^{-3}	1.32×10^{-4}
f_{18}	19.3688	12.8955	103.0825	3.6407	2.8971	1.2763	7.2355	0.2871	6.4787	3.0430	18.5810	2.2303	1.73×10^{-3}	1.09×10^{-3}	6.50×10^{-3}	2.66×10^{-5}

References

1. Qin, C.A. Fast Bayesian Approach for Modal Parameter Identification and Model Updating of Bridge Engineering. Master's Thesis, Hefei University of Technology, Hefei, China, 2020.
2. Mottershead, J.E.; Link, M.; Friswell, M.I. The Sensitivity Method in Finite Element Model Updating: A Tutorial. *Mech. Syst. Signal Process.* **2011**, *25*, 2275–2296. [[CrossRef](#)]
3. Shinozuka, M. Probabilistic Modeling of Concrete Structures. *J. Eng. Mech. Div.* **1972**, *98*, 1433–1451. [[CrossRef](#)]
4. Collins, J.D.; Hart, G.C.; Hasselman, T.K.; Kennedy, B. Statistical Identification of Structures. *Aiaa J.* **1974**, *12*, 185–190. [[CrossRef](#)]
5. Beck, J.L.; Katafygiotis, L.S. Updating Models and Their Uncertainties. I: Bayesian Statistical Framework. *J. Eng. Mech.* **1998**, *124*, 455–461. [[CrossRef](#)]
6. Katafygiotis, L.S.; Beck, J.L. Updating Models and Their Uncertainties. II: Model Identifiability. *J. Eng. Mech.* **1998**, *124*, 463–467. [[CrossRef](#)]
7. Yuen, K.-V.; Beck, J.L.; Katafygiotis, L.S. Efficient Model Updating and Health Monitoring Methodology Using Incomplete Modal Data without Mode Matching. *Struct. Control Health Monit.* **2006**, *13*, 91–107. [[CrossRef](#)]
8. Katafygiotis, L.S.; Papadimitriou, C.; Lam, H.-F. A Probabilistic Approach to Structural Model Updating. *Soil Dyn. Earthq. Eng.* **1998**, *17*, 495–507. [[CrossRef](#)]
9. Yan, W.-J.; Katafygiotis, L.S. A Novel Bayesian Approach for Structural Model Updating Utilizing Statistical Modal Information from Multiple Setups. *Struct. Saf.* **2015**, *52*, 260–271. [[CrossRef](#)]
10. Ching, J.; Beck, J.L. New Bayesian Model Updating Algorithm Applied to A Structural Health Monitoring Benchmark. *Struct. Health Monit.* **2004**, *3*, 313–332. [[CrossRef](#)]
11. Feng, Z.Q.; Katafygiotis, L.S. Bayesian Model Updating Based on Modal Flexibility for Structural Health Monitoring. In Proceedings of the 9th International Conference on Structural Dynamics, Porto, Portugal, 30 June–2 July 2014; pp. 177–184, ISBN 978-972-752-165-4.
12. Li, X.; Tan, J.; Zheng, K.; Zhang, L.; Zhang, L.; He, W.; Huang, P.; Li, H.; Zhang, B.; Chen, Q.; et al. Enhanced Photon Communication through Bayesian Estimation with an Snspsd Array. *Photon. Res.* **2020**, *8*, 637. [[CrossRef](#)]
13. Feng, Z.; Lin, Y. Flutter Derivatives Identification and Uncertainty Quantification for Bridge Decks Based on the Artificial Bee Colony Algorithm and Bootstrap Technique. *Appl. Sci.* **2021**, *11*, 11376. [[CrossRef](#)]
14. Feng, Z.; Ye, Z.; Wang, W.; Lin, Y.; Chen, Z.; Hua, X. Structural Model Identification Using a Modified Electromagnetism-Like Mechanism Algorithm. *Sensors* **2020**, *20*, 4789. [[CrossRef](#)]
15. Fraser, A. Simulation of Genetic Systems by Automatic Digital Computers Ii. Effects of Linkage on Rates of Advance Under Selection. *Aust. J. Biol. Sci.* **1957**, *10*, 492. [[CrossRef](#)]
16. Dorigo, M.; Maniezzo, V.; Colomi, A. Ant System: Optimization by A Colony of Cooperating Agents. *IEEE Trans. Syst. Man Cybern. B* **1996**, *26*, 29–41. [[CrossRef](#)]
17. Yang, X.S. *Nature-Inspired Metaheuristic Algorithms*, 2nd ed.; Luniver Press: England, UK, 2010.
18. Berman, A.; Flannelly, W.G. Theory of Incomplete Models of Dynamic Structures. *Aiaa J.* **1971**, *9*, 1481–1487. [[CrossRef](#)]
19. Feng, Z.; Lin, Y.; Wang, W.; Hua, X.; Chen, Z. Probabilistic Updating of Structural Models for Damage Assessment Using Approximate Bayesian Computation. *Sensors* **2020**, *20*, 3197. [[CrossRef](#)] [[PubMed](#)]
20. O'callahan, J. *System Equivalent Reduction and Expansion Process*; Society of Experimental Mechanics: Bethel, CT, USA, 1989; Volume 7, pp. 29–37.
21. Lal, H.P.; Jith, J.; Gupta, S.; Sarkar, S. Reduced Order Modelling in Stochastically Parametered Acousto-Elastic System Using Arbitrary Pce Based Serep. *Probabilistic Eng. Mech.* **2018**, *52*, 1–14. [[CrossRef](#)]
22. Bansal, S. A New Gibbs Sampling Based Bayesian Model Updating Approach Using Modal Data from Multiple Setups. *Int. J. Uncertain. Quantif.* **2015**, *5*, 361–374. [[CrossRef](#)]
23. Jang, J.; Smyth, A. Bayesian Model Updating of a Full-Scale Finite Element Model With Sensitivity-Based Clustering. *Struct. Control. Health Monit.* **2017**, *24*, 1–15. [[CrossRef](#)]
24. Yang, X.-S. Firefly Algorithm, Lévy Flights and Global Optimization. In *Research and Development in Intelligent Systems Xxvi*; Bramer, M., Ellis, R., Petridis, M., Eds.; Springer: London, UK, 2010; pp. 209–218. ISBN 978-1-84882-982-4.
25. Yu, S.; Su, S.; Lu, Q.; Huang, L. A Novel Wise Step Strategy for Firefly Algorithm. *Int. J. Comput. Math.* **2014**, *91*, 2507–2513. [[CrossRef](#)]
26. Wang, H.; Zhou, X.; Sun, H.; Yu, X.; Zhao, J.; Zhang, H.; Cui, L. Firefly Algorithm with Adaptive Control Parameters. *Soft Comput.* **2017**, *21*, 5091–5102. [[CrossRef](#)]
27. Brajević, I.; Ignjatović, J. An Upgraded Firefly Algorithm With Feasibility-Based Rules for Constrained Engineering Optimization Problems. *J. Intell. Manuf.* **2019**, *30*, 2545–2574. [[CrossRef](#)]
28. Yang, X.-S.; He, X.-S. Why the Firefly Algorithm Works? In *Nature-Inspired Algorithms and Applied Optimization*; Yang, X.-S., Ed.; Studies In Computational Intelligence; Springer International Publishing: Cham, Switzerland, 2018; Volume 744, pp. 245–259. ISBN 978-3-319-67668-5.
29. Yu, S.; Yang, S.; Su, S. Self-Adaptive Step Firefly Algorithm. *J. Appl. Math.* **2013**, *2013*, 1–8. [[CrossRef](#)]
30. Manoharan, G.V.; Shanmugalakshmi, R. Multi-Objective Firefly Algorithm for Multi-Class Gene Selection. *Indian J. Sci. Technol.* **2015**, *8*, 27. [[CrossRef](#)]

31. Wang, C.; Chu, X. An Improved Firefly Algorithm with Specific Probability and Its Engineering Application. *IEEE Access* **2019**, *7*, 57424–57439. [[CrossRef](#)]
32. Shakarami, M.R.; Sedaghati, R. A New Approach for Network Reconfiguration Problem in Order to Deviation Bus Voltage Minimization with Regard to Probabilistic Load Model and Dgs. *Eng. Technol. Int. J. Electr. Comput. Eng.* **2014**, *8*, 430–435.
33. Selvarasu, R.; Surya Kalavathi, M.; Asir Rajan, C.C. Svc Placement for Voltage Constrained Loss Minimization Using Self-Adaptive Firefly Algorithm. *Arch. Electr. Eng.* **2013**, *62*, 649–661. [[CrossRef](#)]
34. Yang, X.S.; He, X. Firefly Algorithm: Recent Advances and Applications. *IJSI* **2013**, *1*, 36. [[CrossRef](#)]
35. Rizk-Allah, R.M.; Zaki, E.M.; El-Saway, A.A. Hybridizing Ant Colony Optimization with Firefly Algorithm for Unconstrained Optimization Problems. *Appl. Math. Comput.* **2013**, *224*, 473–483. [[CrossRef](#)]
36. Yelghi, A.; Köse, C. A Modified Firefly Algorithm for Global Minimum Optimization. *Appl. Soft Comput.* **2018**, *62*, 29–44. [[CrossRef](#)]
37. Brajevic, I.; Ignjatovi, J. An Enhanced Firefly Algorithm for Mixed Variable. *Facta Univ. Ser. Math. Inform.* **2015**, *30*, 401–417.
38. Rajan, A.; Malakar, T. Optimal Reactive Power Dispatch Using Hybrid Nelder–Mead Simplex Based Firefly Algorithm. *Int. J. Electr. Power Energy Syst.* **2015**, *66*, 9–24. [[CrossRef](#)]
39. Ren, Y.; Zhang, L.; Wang, W.; Wang, X.; Lei, Y.; Xue, Y.; Sun, X.; Zhang, W. Genetic-Algorithm-Based Deep Neural Networks for Highly Efficient Photonic Device Design. *Photon. Res.* **2021**, *9*, B247. [[CrossRef](#)]
40. He, W.; Tong, M.; Xu, Z.; Hu, Y.; Cheng, X.; Jiang, T. Ultrafast All-Optical Terahertz Modulation Based on an Inverse-Designed Metasurface. *Photon. Res.* **2021**, *9*, 1099. [[CrossRef](#)]
41. Jamil, M.; Yang, X.S. A Literature Survey of Benchmark Functions for Global Optimisation Problems. *IJMMNO* **2013**, *4*, 150. [[CrossRef](#)]
42. Brajević, I.; Stanimirović, P. An Improved Chaotic Firefly Algorithm for Global Numerical Optimization. *IJCIS* **2018**, *12*, 131. [[CrossRef](#)]