

Article

Improved Adaptive Multi-Objective Particle Swarm Optimization of Sensor Layout for Shape Sensing with Inverse Finite Element Method

Xiaohan Li, Shengtao Niu, Hong Bao * and Naigang Hu

Key Laboratory of Electronic Equipment Structure Design, Ministry of Education, Xidian University, Xi'an 710071, China; xhanli@stu.xidian.edu.cn (X.L.); stniu@stu.xidian.edu.cn (S.N.); nghu@xidian.edu.cn (N.H.)

* Correspondence: hbao@xidian.edu.cn; Tel.: +86-029-882-030-40

Abstract: The inverse finite element method (iFEM) is one of the most effective deformation reconstruction techniques for shape sensing, which is widely applied in structural health monitoring. The distribution of strain sensors affects the reconstruction accuracy of the structure in iFEM. This paper proposes a method to optimize the layout of sensors rationally. Firstly, this paper constructs a dual-objective model based on the accuracy and robustness indexes. Then, an improved adaptive multi-objective particle swarm optimization (IAMOPSO) algorithm is developed for this model, which introduces initialization strategy, the adaptive inertia weight strategy, the guided particle selection strategy and the external candidate solution (ECS) set maintenance strategy to multi-objective particle swarm optimization (MOPSO). Afterwards, the performance of IAMOPSO is verified by comparing with MOPSO applied on the existing inverse beam model. Finally, the IAMOPSO is employed to the deformation reconstruction of complex plate-beam model. The numerical and experimental results demonstrate that the IAMOPSO is an excellent tool for sensor layout in iFEM.



Citation: Li, X.; Niu, S.; Bao, H.; Hu, N. Improved Adaptive Multi-Objective Particle Swarm Optimization of Sensor Layout for Shape Sensing with Inverse Finite Element Method. *Sensors* **2022**, *22*, 5203. <https://doi.org/10.3390/s22145203>

Academic Editor: Theodore E. Matikas

Received: 26 May 2022

Accepted: 10 July 2022

Published: 12 July 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/>).

Keywords: multi-objective particle swarm optimization; external candidate solution set; inverse finite element method; deformation reconstruction; exploitation; exploration

1. Introduction

The deformation of engineering structures under complex working conditions seriously affects the safety and life of the structure. In the past two decades, related scholars in the engineering field have begun to study structural health monitoring (SHM) technology [1–4]. The monitoring of the structure is divided into static monitoring and dynamic monitoring according to the state of the measured structure. Monitoring of static structures has lower requirements on real-time performance, which can be measured directly using measuring equipment, and focuses more on the analysis of the measurement data [5–7]. However, the monitoring of dynamic structures requires higher real-time performance. Structural deformation reconstruction is an important branch of dynamic structural health monitoring. In order to meet the requirements of practical engineering, the deformation reconstruction algorithm should be suitable under various boundary and working conditions.

In Gherlone's analysis on shape sensing of structural deformation [8], the inverse finite element method (iFEM) is an effective deformation reconstruction method. Based on the least square theory, iFEM minimizes the error between surface theoretical strain and actual strain, and then realizes the conversion of surface strain and displacement, which was proposed by Tessler et al. [9,10]. Afterward, Gherlone et al. utilized the mathematical concepts of the iFEM methodology to developed an inverse frame element based on the kinematic assumptions of Timoshenko beam theory, which reconstructed the mode of stretching, bending, transverse-shear, and torsion-deformation [11]. Then, Gherlone et al. numerically and experimentally validated the superior capability of the inverse-frame element for shape sensing of three-dimensional frame structures undergoing static or

dynamic loading [12]. Zhao et al. reconstructed the displacement fields without the knowledge of structural mode shapes, material properties, and applied loading [13]. Niu et al. reconstructed the structural displacements based on strain data collected from the single surface of the structure [14]. Moreover, Chen et al. proposed a unified reconstruction method for the beam-like structure based on the framework of the iFEM, which reduces the structure identification error by adopting some generalized quantities [15].

In the application of the iFEM, some scholars found that the distribution of strain sensors will affect the reconstruction effect. In order to reduce the influence of external environmental factors on the measurement strain, Yuen et al. proposed a Bayesian sequential sensor placement algorithm for multi-type of sensors, based on the robust information entropy [16]. Zhao et al. adopted the particle swarm optimization algorithm to optimize the sensor placement, but it may lead to the degradation of the reconstruction accuracy [17]. Afterwards, Zhao et al. introduced multi-objective particle swarm optimization (MOPSO) to optimize the layout of the sensor, which optimizes both the robustness and the accuracy simultaneously [18]. However, due to the lack of a strategy to jump out of the local optimum, MOPSO is prone to ineffective exploration, with poor effect and low efficiency. The goal of this paper is to propose an optimization algorithm that can be better applied to sensor layout problems.

Optimization of sensor layout in iFEM is a multi-objective optimization problem where model gradient information is not easy to obtain. Traditional methods such as the Conjugate gradient method are no longer applicable. However, with the development of computer technology and biological research, evolutionary algorithms (EA) are gradually used in complex optimization problems. EA is a kind of heuristic random search algorithm based on population evolution, which has the ability to explore the real Pareto front (PF) without gradient information [19]. The key of the multi-objective evolutionary algorithm (MOEA) is that the algorithm achieves a balance between diversity and convergence, so as to effectively generate a set of Pareto optimal solutions that satisfy multiple conflicting objectives at the same time.

The multi-objective particle swarm optimization (MOPSO) [20–22], multi-objective ant colony optimization (MOACO) [23], and multi-objective bee colony optimization (MOBCO) [24,25] are typical multi-objective evolutionary algorithms that are widely used in practical engineering problems. Among them, MOPSO has the advantages of simple implementation and fast convergence speed, which is widely used in the optimization of practical engineering problems [26–28]. However, due to the existence of non-dominated solutions, the quality of the solution cannot be easily assessed, which brings difficulties to the selection of guiding particles. In particular, as the dimension of decision variables gradually increases, the MOPSO can no longer explore effectively.

The current research focuses on how to improve the optimization efficiency on the basis of balancing the convergence and diversity of the algorithm. At present, the improvement for MOPSO is mainly divided into three aspects. On the one hand, the multi-objective problem is solved using a competitive strategy. The competitive particle swarm optimization algorithm introduces a competitive mechanism, which selects the appropriate evolution direction through the game between particles [29–31]. On the other hand, multi-objective problems are optimized through a decomposition strategy. Yu et al. proposed a decomposition-based comprehensive learning particle swarm optimization algorithm, which optimizes each objective in the early stage and cooperates with each other in the later stage [32]. In addition, some scholars improve the guided particle selection strategy to optimize multi-objective problems. Zhu et al. proposed a novel archive-guided velocity update method to guide the transfer of the population [33]. Li et al. introduced a strategy based on ranking dominance to select the elite particles, which can obtain a more diverse and uniform distribution external archive [34]. The above algorithms introduce strategies on MOPSO to improve the performance of the algorithm, but as the target dimension increases, a single strategy will make the algorithm enter into long-term ineffective exploration, or

even fall into local optimum. The IAMOPSO algorithm proposed in this paper focuses not only on convergence and diversity, but also on the efficiency of exploration.

To evaluate the effect of reconstruction, not only the accuracy, but also the robustness (the ability to resist external interference when the sensor is installed) should be considered. In this paper, the reconstruction accuracy is evaluated using the root mean square error (RMSE) of the actual displacement and theoretical displacement, and the robustness is evaluated by the composite sensitivity matrix.

This paper is organized as follows. Section 2 briefly introduces the inverse finite element method for beam elements, and then establishes an optimization model of sensor distribution based on reconstruction accuracy and stability metrics. Section 3 proposes an improved adaptive multi-objective particle swarm optimization (IAMOPSO) algorithm. Section 4 applies IAMOPSO to the sensor layout problem on the deformation reconstruction of the circular beam model. The numerical and experimental verification results confirm that IAMOPSO is an excellent tool for sensor placement in iFEM.

2. Review of Beam Deformation Reconstruction through iFEM

2.1. Inverse Finite Element Method

The inverse finite element method (iFEM) is described in detail in Ref. [11]. The principle is briefly described in this section. As shown in Figure 1, for a Timoshenko beam, the deformation of the neutral axis can be represented by the displacement along each axis and the rotation angle around each axis. The displacement of the neutral axis \mathbf{u} can be expressed as Equation (1).

$$\mathbf{u} = [u, v, w, \theta_x, \theta_y, \theta_z]^T \quad (1)$$

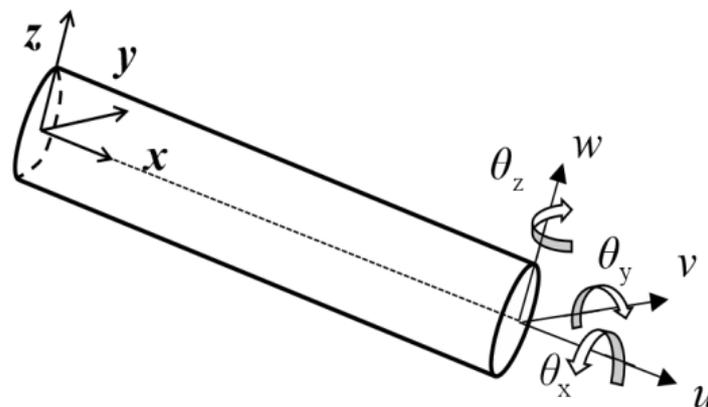


Figure 1. Schematic diagram of Timoshenko beam.

The displacement of any point on the beam cross-section along the direction x , direction y , and direction z can be represented by the neutral axis displacement:

$$\begin{aligned} u_x(x, y, z) &= u(x) + z\theta_y(x) - y\theta_z(x) \\ u_y(x, y, z) &= v(x) - z\theta_x(x) \\ u_z(x, y, z) &= w(x) + y\theta_x(x) \end{aligned} \quad (2)$$

Define arbitrary section strain as: $\mathbf{e}(\mathbf{u}) = [e_1, e_2, e_3, e_4, e_5, e_6]^T$. The strain along the axis at any point is the derivation of Equation (2) with respect to x :

$$\begin{aligned} \varepsilon_x(x, y, z) &= e_1(x) + ze_2(x) + ye_3(x) \\ \gamma_{xz}(x, y) &= e_4(x) + ye_6(x) \\ \gamma_{xy}(x, y) &= e_5(x) - ze_6(x) \end{aligned} \quad (3)$$

The section strain can be described as Equation (4).

$$\begin{aligned} e_1(x) &= u_x(x), e_2(x) = \theta_{y,x}(x), e_3(x) = -\theta_{z,x}(x) \\ e_4(x) &= w_x(x) + \theta_y(x), e_5(x) = v_x(x) - \theta_z(x), e_6(x) = \theta_{x,x}(x) \end{aligned} \quad (4)$$

Based on the least square method, iFEM minimizes the theoretical strain and experimental strain errors. The fitting function $\phi(\mathbf{u})$ between the theoretical section strain $e(\mathbf{u})$ and the actual section strain e^ε is defined as follows:

$$\phi(\mathbf{u}) = \|\mathbf{e}(\mathbf{u}) - e^\varepsilon\|^2 \quad (5)$$

The displacement at any section position can be represented by the shape function $N(x)$ and the nodal degrees of freedom \mathbf{u}_e :

$$\mathbf{u}(x) = N(x)\mathbf{u}_e \quad (6)$$

After that, the strain at any section position can be calculated by Equation (7).

$$\mathbf{e}(\mathbf{u}) = \mathbf{B}(x)\mathbf{u}_e \quad (7)$$

where $\mathbf{B}(x)$ is called the strain function matrix, which is the derivative of $N(x)$. In order to minimize the error of theoretical strain and actual strain, let the derivative function of Equation (5) with respect to \mathbf{u} be zero, then Equation (8) can be obtained.

$$\mathbf{k}^e \mathbf{u}_e = \mathbf{f}^e \quad (8)$$

\mathbf{k}^e and \mathbf{f}^e can be expressed as:

$$\begin{aligned} \mathbf{k}^e &= \sum_{k=1}^6 w_k \mathbf{k}_k^e & \mathbf{k}_k^e &= \frac{L}{n} \sum_{i=1}^n [\mathbf{B}_k^T(x_i) \mathbf{B}_k(x_i)] \\ \mathbf{f}^e &= \sum_{k=1}^6 w_k \mathbf{f}_k^e & \mathbf{f}_k^e &= \frac{L}{n} \sum_{i=1}^n [\mathbf{B}_k^T(x_i) e_k^\varepsilon(x_i)] \end{aligned} \quad (9)$$

where L denotes the element length, n is the number of section strain distributions on the neutral axis, and x_i represents the location where the section strains are evaluated; w_k is a weighting factor considering the mutual influence of the axial stretching, bending, twisting, and transverse shearing; $e_k^\varepsilon(x_i)$ are the section strains at the location x_i , which is computed from strain-sensor values ε_2^* , shown in Ref. [11]. \mathbf{k}^e is a function of x_i , whereas \mathbf{f}^e depends on x_i and the measured strain values ε_2^* .

After splicing multiple inverse finite elements, Equation (10) can be obtained.

$$\mathbf{K}\mathbf{U} = \mathbf{F} \quad (10)$$

where \mathbf{K} is global stiffness matrix, which is related to section position x_i ; \mathbf{F} is the global load vector, which depends on x_i and ε_2^* . The matrix-vector equation provides a solution for the nodal freedom of structural deformation, $\mathbf{U} = \mathbf{K}^{-1}\mathbf{F}$. Then, through Equation (6), the deformation of any section position can be obtained by interpolation. According to Equation (2), the deformation of any point can be calculated eventually.

2.2. Indexes to Evaluate the Effect of Deformation Reconstruction

After reconstructing the displacement field of the structure based on the iFEM, this section proposes two indicators to evaluate the effect of the reconstruction: reconstruction accuracy and robustness metrics. The former estimates the accuracy of reconstruction, and the latter reflects the stability of the reconstructed model.

When the sensor layout scheme \mathbf{X} is selected, the reconstruction accuracy index $RRMSE(\mathbf{X})$ is defined as follows:

$$RRMSE(\mathbf{X}) = \frac{RMSE(\mathbf{X})}{\max(\text{disp}^F)} \times 100\% \quad (11)$$

$$RMSE(\mathbf{X}) = \sqrt{\frac{1}{N} \sum_{i=1}^N (disp_i^F(\mathbf{X}) - disp_i^P(\mathbf{X}))^2} \quad (12)$$

where $disp^F$ represents the theoretical displacement, which refers to the deformation value from simulation; $disp^P$ represents the displacement obtained by deformation reconstruction; N is the number of check points.

To explicitly the influence of strain–sensor values on reconstruction stability, Equation (10) can be transformed as

$$\mathbf{U} = \mathbf{A}(\mathbf{X})\varepsilon_2^* \quad (13)$$

where $\mathbf{A}(\mathbf{X})$ is referred as a strain–deformation converted matrix, which is only related to the location of sensors; ε_2^* is the measurement strain vector, and \mathbf{U} is the nodal freedom of structural deformation.

The method proposed in Ref. [35] is used to measure the reconstruction stability, as shown in Equation (14).

$$\cos(\theta_i(\mathbf{X})) = \frac{\mathbf{A}_i(\mathbf{X})\mathbf{P}(\mathbf{X})\mathbf{A}_i^T(\mathbf{X})}{\|\mathbf{A}_i(\mathbf{X})\| \times \|\mathbf{P}(\mathbf{X})\mathbf{A}_i^T(\mathbf{X})\|} \quad (14)$$

$$\mathbf{P}_i(\mathbf{X}) = \overline{\mathbf{A}}_i^T(\mathbf{X})(\overline{\mathbf{A}}_i(\mathbf{X})\overline{\mathbf{A}}_i^T(\mathbf{X}))^{-1}\overline{\mathbf{A}}_i(\mathbf{X}) \quad (15)$$

where $\mathbf{A}_i(\mathbf{X})$ is the i -th row vector of matrix $\mathbf{A}(\mathbf{X})$, and $\overline{\mathbf{A}}_i(\mathbf{X})$ consists of the remaining row vectors of $\mathbf{A}(\mathbf{X})$ minus $\mathbf{A}_i(\mathbf{X})$. $\theta_i(\mathbf{X})$ is calculated for each row vector, and the robustness index selects its maximum value. The reconstruction robustness index is defined as

$$RBI(\mathbf{X}) = \max(\theta_i(\mathbf{X})) \quad i = (1, 2, \dots, m) \quad (16)$$

where m denotes the dimension of the structural node degrees of freedom. The bigger the value of $RBI(\mathbf{X})$, the higher the error tolerance of matrix $\mathbf{A}(\mathbf{X})$ to sensor errors.

The purpose of the sensor layout problem in iFEM is to find a sensor scheme that achieves a balance between reconstruction accuracy and reconstruction stability. The optimization model of sensor distribution is described as follows:

$$\begin{aligned} \text{Minimize Index}(\mathbf{X}) &= [RRMSE(\mathbf{X}), -RBI(\mathbf{X})] \\ \mathbf{X} &= (X_1, X_2, \dots, X_6) \end{aligned} \quad (17)$$

where the number of sensors used in the reconstruction method is six; $Index(\mathbf{X})$ represents objective function, consists of accuracy index $RRMSE(\mathbf{X})$ and robustness index $-RBI(\mathbf{X})$. The IAMOPSO proposed in Section 3 is used to find sensor schemes that minimize both metrics simultaneously.

3. Optimization Algorithm

3.1. Multi-Objective Particle Swarm Optimization Algorithm

The Particle Swarm Optimization (PSO) algorithm is a swarm intelligence optimization algorithm, which was proposed by Kennedy and Eberhart [36]. It guides individuals to transfer through the coordination among individuals, as shown in Equation (18) [37].

$$\mathbf{X}_i(t+1) = \mathbf{X}_i(t) + \mathbf{V}_i(t+1) \quad (18)$$

$$\mathbf{V}_i(t+1) = w \cdot \mathbf{V}_i(t) + c_1 \cdot r_1 \cdot (\mathbf{pb}_i(t) - \mathbf{X}_i(t)) + c_2 \cdot r_2 \cdot (\mathbf{gb}_i(t) - \mathbf{X}_i(t)) \quad (19)$$

where $\mathbf{pb}_i(t)$ is the individual optimal, which represents the optimal position of particle $\mathbf{X}_i(t)$ in the past generation. $\mathbf{gb}_i(t)$ is the group optimal, which represents the optimal location explored by all individuals in the group. w is the inertia weight (all w in Section 3 represent inertia weight). c_1 and c_2 are acceleration factors. r_1 and r_2 are random values between 0 and 1. The velocity at the next moment $\mathbf{V}_i(t+1)$ of the particle $\mathbf{X}_i(t)$ is deter-

mined by the current velocity $V_i(t)$, individual optimal $pb_i(t)$, and group optimal $gb_i(t)$. The particle moves from $X_i(t)$ to $X_i(t+1)$ with an updated velocity $V_i(t+1)$, as shown in Figure 2.

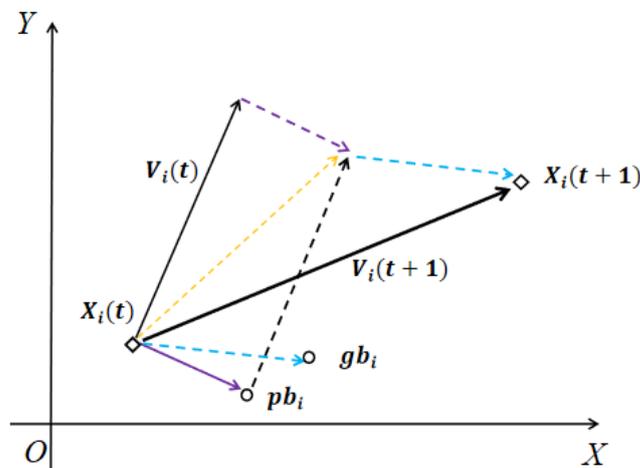


Figure 2. The transfer process of particle swarm optimization.

In multi-objective problems (MOP), there may be multiple exclusive objective functions, which make the quality of the solutions impossible to be simply compared. The Pareto dominance relation is defined as follows: If for X and Y in the decision space, $\forall j \in \{1, 2, \dots, m\}, f_j(X) \leq f_j(Y)$, and $\exists j \in \{1, 2, \dots, m\}, f_j(X) < f_j(Y)$, m represents the number of objective functions, then the decision variable X is defined to dominate the decision variable Y . The final solution of MOP is not a single solution, but a set of non-dominated solutions.

Throughout the exploration, for MOP, the following issues need to be considered:

1. How to guide particles to converge:

Algorithm convergence is the foundation of an optimization algorithm. As shown in Figure 2, the global optimum gb_i and the individual optimum pb_i influence the iterative direction in each generation; hence, it is crucial to adopt an appropriate guided particle selection strategy.

2. How to balance the exploration and exploitation of algorithm:

If the algorithm focuses on exploitation and ignores exploration, particles may converge rapidly and fall into a local optimum. If the algorithm focuses on exploration and neglects exploitation, the algorithm may explore ineffectively.

3. How to search the target space fully:

For problems with complex shapes of the Pareto front, the algorithm needs to fully explore the target space to ensure the integrity of the Pareto front.

4. How to improve exploration efficiency in a high-dimensional space:

Algorithms need to explore the Pareto front in a finite time, so the particles should explore as efficiently as possible. In the guidance process, the particle information needs to be fully utilized to help the particles make better transfer strategies.

3.2. Improved Adaptive Multi-Objective Particle Swarm Optimization Algorithm

In this section, in order to balance the convergence and development of the algorithm and improve the efficiency of the algorithm, the IAMOPSO introduces the following four strategies on the basis of MOPSO. Moreover, the strategies employ external candidate solution (ECS) set consisting of non-dominated solutions obtained by the dominance relationship, which is updated after each particle transfer.

3.2.1. Initialization Strategy

MOPSO usually adopts the method of uniform initialization to generate the initial population. When the variable dimension is high and the target space is large, the population needs to explore for a long time. In addition, there are few particles in the ECS set in the early stage, which tends to trap the population in local optima. To solve this problem, an initialization method based on a single-objective problem is proposed in this paper.

Before formally entering the multi-objective optimization with m objectives, m times of single-objective optimization are carried out, respectively. In the single-objective optimization, the population size generally takes $N_{\text{Ini}} < N/5$ and the number of iterations takes $T_{\text{Ini}} < T/10$, where N is the number of particles and T is the maximum number of iterations. Some particles of each generation retained in the single-objective optimization process, which constitute the initial population $P = P_{\text{IniF1}} \cup P_{\text{IniF2}} \cdots \cup P_{\text{IniFm}}$, where $P_{\text{Ini-Fi}}$ represents the population composed of particles retained by single-objective optimization for the i -th objective. Take $m = 2$ as an example, Figure 3 shows the composition of the initial population for dual-objective optimization. The red particles are the particles retained in the process of optimizing the objective function F1, while the blue particles are the particles retained in the process of optimizing the objective function, F2. This is because, in the whole optimization process of single-objective optimization, each particle tends to cluster around the optimal position of the objective. Using the initialization strategy proposed in this section to synthesize a multi-objective initial population can maintain this trend and produce initial solutions with better performance.

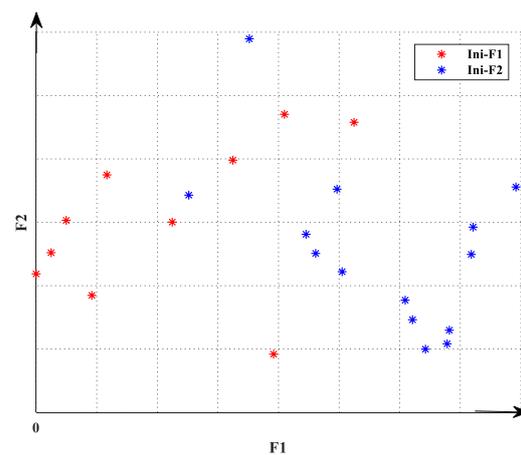


Figure 3. The composition of the initial population in dual-objective optimization.

3.2.2. Adaptive Inertia Weight Strategy

The adaptive inertia weight strategy makes the algorithm focus on exploration ability in the early stage and exploitation ability in the later stage. It can be observed from the Equation (19) that the algorithm has better exploration ability when $w > 1$, and gradually converges when $w < 1$. IAMOPSO introduces a time factor to the inertia weight w , and maps the Sigmoid function to the curve of w .

$$w(t) = w_{\max} - (w_{\max} - w_{\min}) \cdot \text{sigmoid}\left(\frac{\Delta S \times t}{T} - S_{\max}\right) \quad (20)$$

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)} \quad (21)$$

$$\Delta S = S_{\max} - S_{\min} \quad (22)$$

where T represents the maximum number of iterations.

Taking different S_{\min} and S_{\max} can dynamically adjust w , so as to control the exploration and exploitation of the algorithm. As shown in Figure 4, w_{\max} takes 1.2 and w_{\min}

takes 0.8. The three curves in the figure use (S_{\min}, S_{\max}) as $(-6, 6)$, $(-4, 6)$, $(-6, 4)$, which map out different w and control the transition of the algorithm.

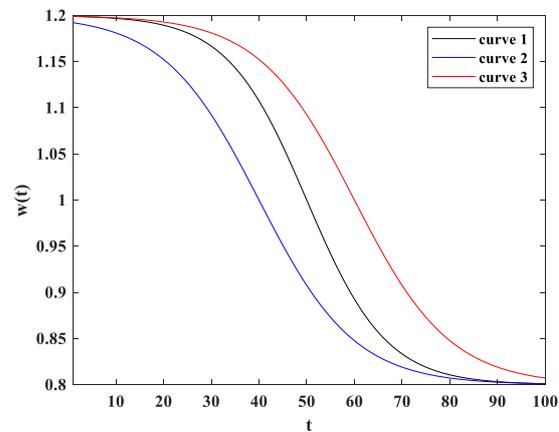


Figure 4. Adaptive process of inertial weight.

3.2.3. Guided Particle Selection Strategy

In order to improve the efficiency of IAMOPSO, this paper adopts the following adaptive method to select the guiding particle: When ECS is improved, the population maintains the state of exploitation, and selects the particle closest to the origin in ECS set as gb . when the ECS is not improved in a long time, the population enters the state of exploration, and selects an individual in the sparse region of the ECS set as gb . The ECS set is partitioned by angle in each generation, as shown in Figure 5. When the population is in an exploitation state, the algorithm calculates the Euclidean distance between each point and the origin. The distance between particle A and the origin is the smallest, so particle A is selected as gb . When the population is in exploration state, particle A can no longer effectively guide particles, while particle B in the sparse region can guide particles to explore a uniformly distributed Pareto front. Particle B is selected as gb in this iteration.

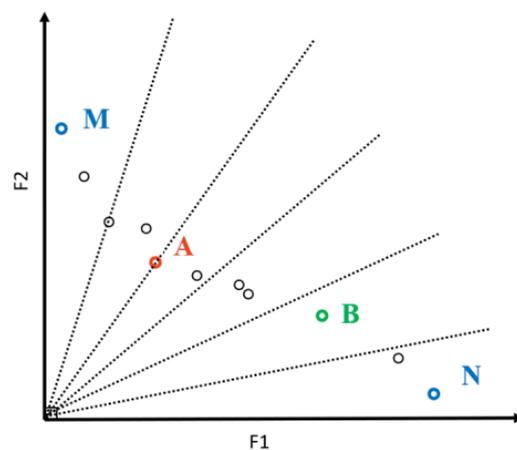


Figure 5. Globally optimal particles in different evolution states.

Since different objective functions have different scales in practical applications, it is necessary to normalize the values of each objective function before partitioning. The maximum and minimum values of particles in the ECS set are used for normalization, as in Equation (23). $F_j(X_i)$ is the actual target value, and $F'_j(X_i)$ is the value after normalization of the particle X_i . As shown in Figure 5, the particle M corresponds to F_1^{\min} and F_2^{\max} , while the particle N corresponds to F_1^{\max} and F_2^{\min} . In this way, the particles in the ECS

set are normalized into a unit hypercube of $[0, 1]$, which facilitates the estimation of the evolutionary state.

$$\begin{aligned} F'_j(\mathbf{X}_i) &= (F_j(\mathbf{X}_i) - F_j^{\min}) / (F_j^{\max} - F_j^{\min}) \\ F_j^{\max} &= \max_{\mathbf{x}_i \in P_{\text{ECS}}} F_j(\mathbf{X}_i) \\ F_j^{\min} &= \min_{\mathbf{x}_i \in P_{\text{ECS}}} F_j(\mathbf{X}_i) \end{aligned} \quad (23)$$

In the early stage of the iterative algorithm, there are few solutions in the ECS. If there are many partitions, many continuous sparse regions will be generated, resulting in low efficiency. In the later stage of the iterative algorithm, the number of solutions in the ECS set is large. If there are too few partitions, the sparse regions cannot be effectively identified during partitioning. In order to solve the above problems, the algorithm adopts an adaptive partition strategy according to the number of particles in the ECS set, which guides the population to explore sparse regions more efficiently.

3.2.4. External Candidate Solution Set Maintenance Strategy

In the later stage of the iterative algorithm, the number of particles in ECS set explodes, resulting in too much computational time. To reduce the selection pressure, some redundant particles need to be dynamically removed when the number of particles in ECS set reaches the threshold. The maintenance of non-dominated solutions focuses on how to reduce the selection pressure while maintaining the shape of PF.

Here, we use the partitioning strategy proposed in Section 3.2.3. We specify the maximum capacity of the ECS set N_{MaxArc} . After each iteration, the ECS set is updated according to the Pareto dominance relationship. If the number of particles in the ECS set $N_{\text{Arc}} > N_{\text{MaxArc}}$, the algorithm maintains the external candidate solution. The algorithm first calculates the number of redundant particles, and then divides the ECS set into D regions: $A = \{A_1, A_2, \dots, A_D\}$. Select a set of dense regions A_{dense} , and randomly delete particles in dense region A_{dense} according to the proportion.

A dense region is defined as: if the number of particles in the region k satisfy $\rho(A_k) - \rho(A_{\text{sparse}}) \geq N_{\text{remove}}$, where $A_{\text{sparse}} = \min_{A_i} \rho(A_i), i \in [1, D]$, then A_k belongs to A_{dense} where $\rho(A_i)$ is the particle density (number of particles) in region A_i .

As shown in Figure 6, the maximum capacity of ECS N_{MaxArc} is set to 20, while the current number of particles in ECS N_{Arc} is 25. At this time, redundant particles need to be dynamically removed, and the number of redundant particles N_{remove} is 5. Calculating the particle density in each region $\rho(A_i)$, it can be found $A_{\text{sparse}} = \min_{A_i} \rho(A_i) = A_5$, and $\rho(A_{\text{sparse}}) = \rho(A_5) = 1$. Since $\rho(A_1) = 9$ and $\rho(A_3) = 6$, which satisfy $\rho(A_k) - \rho(A_{\text{sparse}}) \geq N_{\text{remove}}$, then A_1 and A_3 belong to the dense region. Three redundant particles in A_1 and two redundant particles in A_3 are randomly removed according to the proportion (the red particles in Figure 6 will be dynamically deleted).

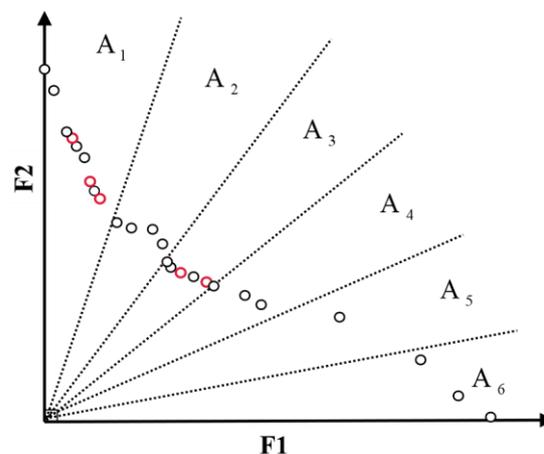


Figure 6. Maintenance of external candidate solution set.

3.2.5. Algorithm Framework

The general framework of IAMOPSO is described as follows (see Algorithm 1).

Algorithm 1: Framework of IAMOPSO

Input: Initialized population size N_{Ini} , initialized number of iterations T_{Ini} , number of particles N , maximum number of iterations T , maximum capacity of ECS set N_{MaxArc} , adaptive inertia weight parameters w_{max} , w_{min} , S_{min} , and S_{max} , and number of initial partitions $D = 3$

Termination condition: The maximum number of iterations is reached.

Step 1: Initialize

1. Use the initialization strategy to generate the initial population.
2. Select the non-dominated solutions in the initial population and store them in the ECS set.

Step 2: Iteration

1. The inertia weight w is adaptive.
2. Estimate the evolutionary state. At the beginning of the iteration, the population is in the exploitation state.
3. Partition the target space. Calculate the population density in each partition.
4. Select gb according to the evolution state.
5. Calculate V and perform particle transfer.
6. Update ECS set to ensure that all non-dominated solutions explored are stored in the ECS set.
7. If the number of external candidate solutions exceeds the limit, maintain ECS set and remove redundant particles.
8. Determine whether the termination condition is reached, if it is satisfied, output the ECS set, otherwise enter the next iteration.

Output: ECS set

3.3. Technical Route

The sensor layout optimization employing the proposed IAMOPSO algorithm comprises three steps, which are illustrated in the schematic diagram of Figure 7. After the model of sensor distribution is established, the accuracy index $RRMSE(\mathbf{X})$ and robustness index $RBI(\mathbf{X})$ of sensor scheme are used as fitness functions in the IAMOPSO algorithm to evaluate the quality of each scheme. After optimization with IAMOPSO, a set of non-dominated sensor schemes are obtained. Then, the evaluation of IAMOPSO is achieved through numerical and experimental evaluation on circular beams and complex models.

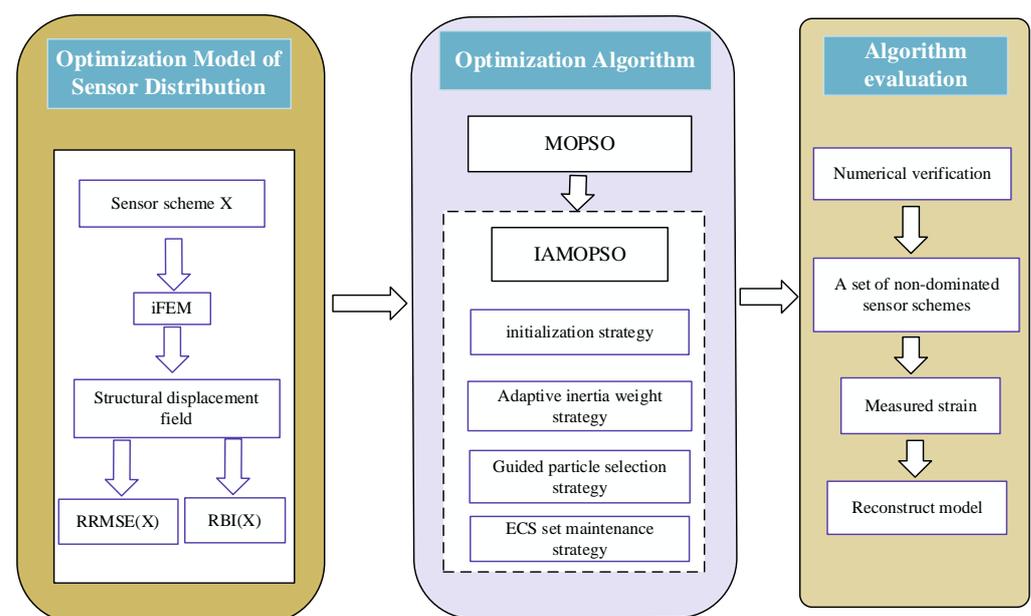


Figure 7. Technical route for sensor layout optimization employing IAMOPSO.

4. Algorithm Evaluation

4.1. Numerical Examples

In this section, to evaluate the effect of the IAMOPSO algorithm on the sensor layout in the iFEM, a cantilever beam is reconsidered for the simulation example compared with the MOPSO proposed in Ref. [18]. The model employs a beam of homogeneous solid with the engineering simulation finite element analysis, which is divided into 200 elements and the cross-section of the beam is divided into 36 segments, as shown in Figure 8. The length of the beam model is $L = 660$ mm, the outer radius is $R = 13$ mm, and the inner radius is $r = 11.5$ mm. The target space is a series of position points, which carry strain information. The Young's modulus is $E = 7.03$ GPa, the Poisson's ratio is $\nu = 0.3$, and the density is $\rho = 2700$ kg/m³.

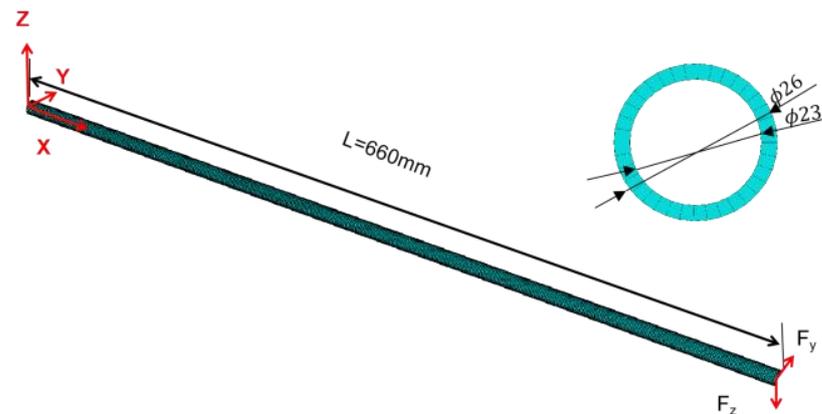


Figure 8. Finite element model of the beam.

Different loads are subjected to the free end of cantilever beam, which are formed by the simultaneous application of two loads, as shown in Table 1.

Table 1. Load applied at the end of the beam.

Load	Heavy Load (N)	Moderate Load (N)
F_y	250	120
F_z	200	100

Six sensors (a sensor distribution scheme) were used to reconstruct the displacement field of the beam, and 200 checkpoints were selected to calculate the *RRMSE*. Based on the optimization model of sensor distribution proposed in Section 2, the IAMOPSO algorithm is used to optimize the layout of sensor, while the MOPSO proposed in Ref. [17] is used to compare and verify the performance of IAMOPSO. All sensor distribution schemes obtained and the Pareto front with IAMOPSO are shown in Figure 9.

As shown in Figure 9b, the distribution of the Pareto front shows that the reconstruction effect with IAMOPSO exceeds with MOPSO significantly. Moreover, by observing the particle transfer in the optimization process, it can be found that the MOPSO lacks the ability to jump out of the local optimum, which is inefficient. Meanwhile IAMOPSO can make full use of the information of particles and jump out of the local optimum, which is efficient.

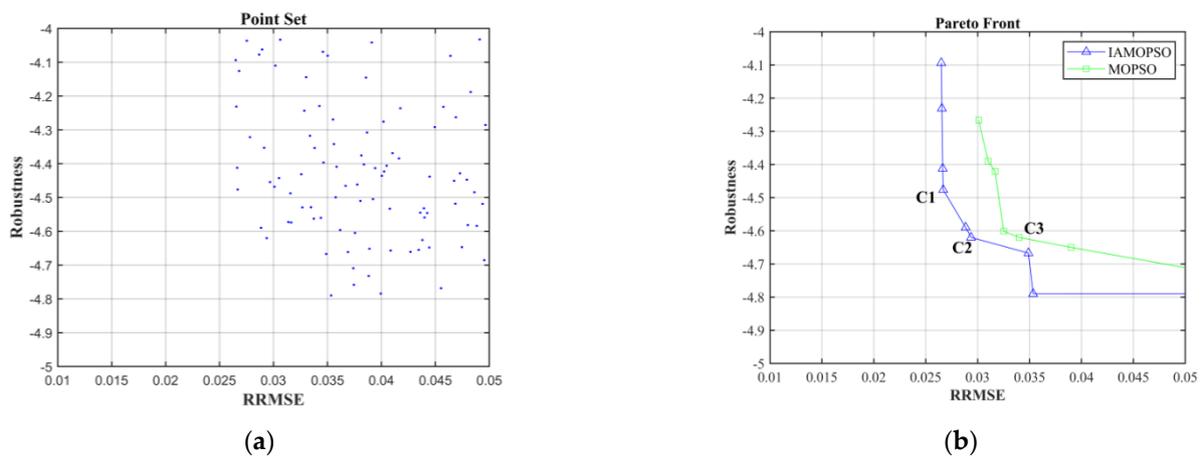


Figure 9. Optimization results: (a) The objective function corresponding to the sensor distribution scheme explored in the IAMOPSO optimization process; (b) comparison of Pareto front explored by IAMOPSO and MOPSO.

The typical positions C1 and C2 on the Pareto front are selected for comparison with the position C3 obtained by the MOPSO algorithm, and the position information is shown in Table 2. In order to intuitively measure the reconstruction effect, the maximum error (ERR^{\max}) is introduced, which is defined as Equation (24).

$$ERR^{\max} = \max |disp_i^F(\mathbf{X}) - disp_i^P(\mathbf{X})|, i = 1, \dots, 200 \quad (24)$$

where $disp_i^F(\mathbf{X})$ represents the theoretical displacement of i -th check points and $disp_i^P(\mathbf{X})$ represents the displacement of the i -th check point obtained by deformation reconstruction.

Table 2. The distribution scheme of the sensors with IAMOPSO (C1, C2) and MOPSO (C3) (mm).

	C1	C2	C3
ε_1	(190.00, −10.32, −10.00)	(40.00, −10.32, −10.00)	(217.00, −10.32, 10.00)
ε_2	(257.00, 13.00, 0.00)	(168.00, −12.69, 3.47)	(343.00, −11.79, 6.84)
ε_3	(182.00, −3.47, 12.69)	(318.00, 3.47, −12.69)	(212.00, −8.33, 10.86)
ε_4	(408.00, −8.33, −10.86)	(288.00, 0.00, 13.00)	(328.00, −10.32, 10.00)
ε_5	(123.00, 11.79, −6.84)	(38.00, −12.69, 3.47)	(252.00, −12.69, −3.47)
ε_6	(348.00, −3.47, 12.69)	(248.00, 10.86, 8.33)	(248.00, 6.84, −11.79)

Table 3 compares the ERR^{\max} of the three typical schemes without interference, where Max_disp represents the maximum theoretical deformation displacement. The result shows that the reconstruction accuracy of the schemes C1 and C2 are better than that of the scheme C3.

Table 3. Comparison of ERR^{\max} under three sensor distribution schemes (mm).

Load (N)	Direction	Max_Displacement	C1	C2	C3
Heavy Load	X	0	6.2×10^{-4}	6.8×10^{-4}	7.9×10^{-4}
	Y	39.2	0.07	0.10	0.19
	Z	31.5	0.05	0.06	0.08
Middle Load	X	0	5.5×10^{-4}	6.3×10^{-4}	5.8×10^{-4}
	Y	18.9	0.06	0.08	0.11
	Z	16.3	0.06	0.07	0.09

In order to verify the robustness, interferences are applied to three schemes, as shown in Equation (25). When the sensor distribution scheme is determined, these interferences

randomly distributed in the error range are added 500 times on each sensor independently, and the maximum error $RRMSE^{err}$ is recorded.

$$\Delta x_i \in (-0.3, 0.3), \Delta y_i \in (-0.05, 0.05), \Delta z_i \in (-0.05, 0.05), i = 1, \dots, 6 \quad (25)$$

Table 4 shows the values of robustness and accuracy index on three schemes. The $RRMSE$ of the three schemes is 2.6%, 2.8% and 3.4% without interference, while they degrade to 6.9%, 5.1% and 11.3% with interference applied, respectively. Obviously, C2 has excellent resistance to interference, which is more suitable for complex working conditions.

Table 4. Comparison of reconstruction accuracy and robustness metrics.

Optimizer	Scheme	$RRMSE$	Robustness Index	$RRMSE^{err}$
IAMOPSO	C1	0.026	−4.41	0.069
	C2	0.028	−4.62	0.051
MOPSO	C3	0.034	−4.61	0.113

In conclusion, according to the above analysis results, the sensor scheme C2 dominates C3 on all evaluation metrics, which indicates that the optimization effect of IAMOPSO obviously exceeds that of MOPSO.

4.2. Numerical and Experimental Validation

In this section, to demonstrate the universality of IAMOPSO algorithm for sensor layout applications, numerical and experimental verifications are performed on a complex plate-beam structure, which consists of three beams and one plate. The length of the beam is $L_c = 2200$ mm, the outer radius is $R = 20$ mm, and the inner radius is $r = 16$ mm. The length of the board is 1500 mm, the width is 400 mm, and the thickness is 8 mm. The board is 470 mm from the fixed end. The distance between the main beam (middle beam) and the two symmetrical sub-beams is 130 mm. The ribs are 400 mm long, 50 mm wide, and 20 mm thick. The distances from the six ribs to the fixed end are 250 mm, 520 mm, 870 mm, 1220 mm, 1570 mm, and 1920 mm, respectively, as shown in Figure 10. The Young's modulus is $E = 7.03$ GPa, the Poisson's ratio is $\nu = 0.3$, and the density is $\rho = 2700$ kg/m³. In the simulation, a static load of 350 N was applied to the free end of the beam.

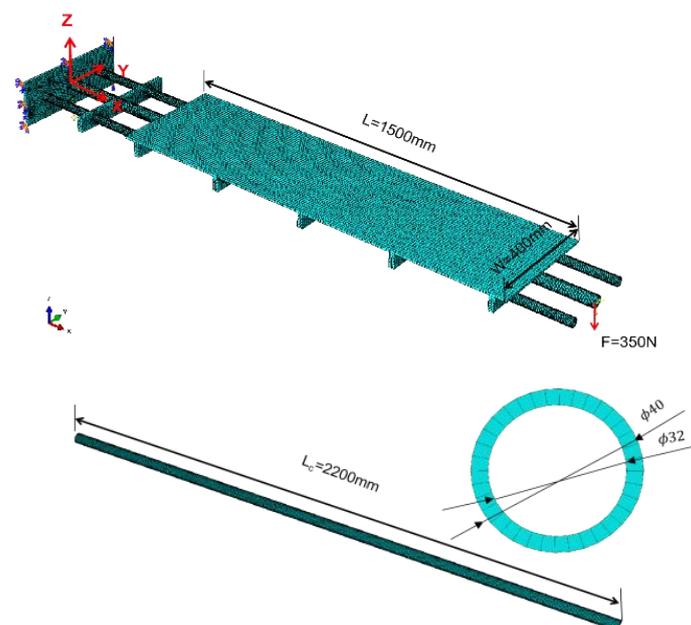


Figure 10. Complex plate-beam structure under concentrated load.

Considering robustness and accuracy as indexes, the sensor distribution model is established. Then, the IAMOPSO algorithm is used to optimize the sensor placement, and the Pareto front is shown in Figure 11. Non-dominated schemes $P_1 \sim P_4$ on PF are selected for detailed description, whose location information is shown in Table 5.

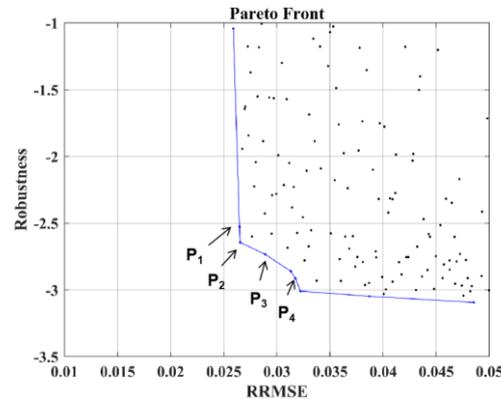


Figure 11. Pareto front based on IAMOPSO.

Table 5. Sensor location information of scheme P_1 to P_4 (mm).

	P_1	P_2	P_3	P_4
ε_1	(485.00, 6.84, -18.79)	(955.00, -6.84, -18.79)	(840.00, 19.69, -3.47)	(890.00, 15.32, 12.85)
ε_2	(640.00, 6.84, -18.79)	(415.00, -0.00, -20.00)	(550.00, 17.32, 10.00)	(695.00, 10.00, -17.32)
ε_3	(785.00, 3.47, -19.69)	(880.00, -10.00, -17.32)	(380.00, 3.47, -19.69)	(475.00, 3.47, 19.69)
ε_4	(745.00, 10.00, 17.32)	(565.00, 18.79, -6.84)	(545.00, 18.79, 6.84)	(620.00, 17.32, 10.00)
ε_5	(625.00, 19.69, 3.47)	(645.00, 6.84, -18.79)	(1080.00, 19.69, 3.47)	(965.00, 15.32, 12.85)
ε_6	(1660.00, -20.00, 0.00)	(495.00, -10.00, -17.32)	(835.00, -17.32, 10.00)	(750.00, 10.00, -17.32)

Figure 12 shows the corresponding sensor location of the four schemes $P_1 \sim P_4$ in the model. It can be observed that a position of scheme P_2 is relatively close to the rib, hence scheme P_2 is not given priority considering the difficulty of installation. The remaining layout schemes are finally selected according to the actual engineering needs.

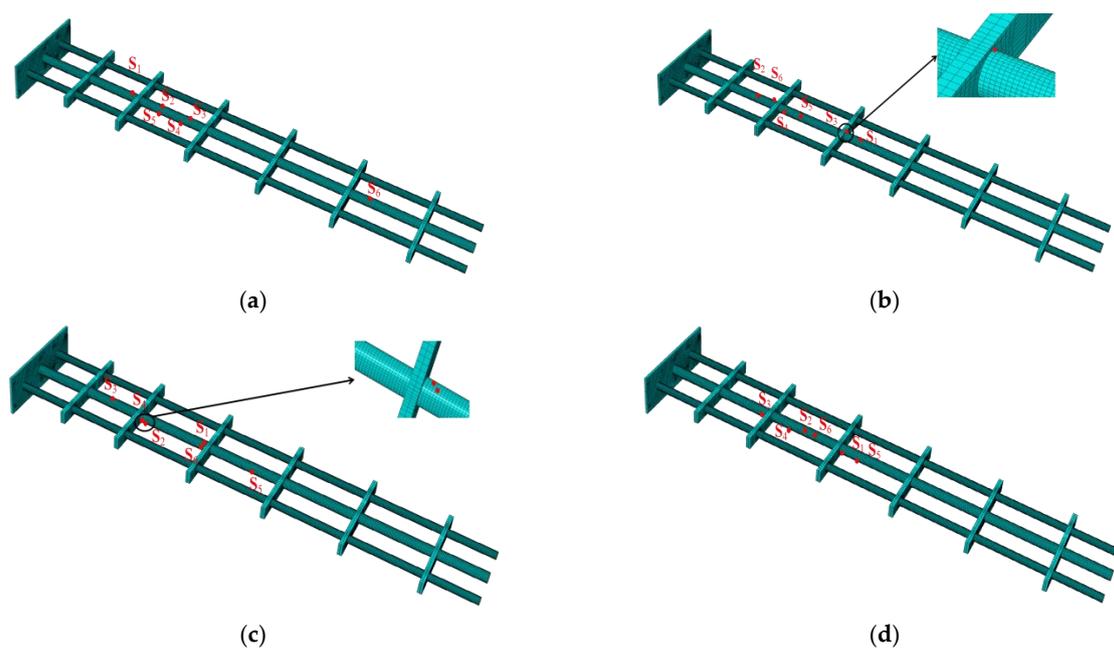


Figure 12. The position corresponding to P_1 (a), P_2 (b), P_3 (c), and P_4 (d) in the model.

In the experiments, two types of sensors are used, one is the fiber Bragg grating (FBG) sensor that captures the surface strain, and the other is the position sensor that captures the true displacement of the marked points (check points). According to the sensor distribution scheme obtained by simulation, six fiber FBG sensors are installed on the main beam to reconstruct the displacement field, and position sensors are installed at the marked points of the plate, the positions of which are captured by a 3D optical measurement device (NDI Optotrak Certus, Northern Digital Incorporation, Waterloo, ON, Canada), as shown in Figure 13.



Figure 13. Experimental platform: (a) Loading on end-node of entire structure; (b) NDI Optotrak Certus.

Remove the unreasonable sensor scheme P_2 , and select the sensor schemes P_1 and P_3 with higher accuracy for experimental tests with different end node loads in the Z direction. The accuracy analysis is shown in Table 6, and $iFEM^{\max}$ represents the maximum reconstructed deformation, while NDI^{\max} represents maximum experimental deformation measured by NDI.

Table 6. Analysis of reconstruction accuracy under two concentrated loads.

Index	200 N	350 N
NDI^{\max}	−100.29 mm	−144.09 mm
$iFEM^{P1_{\max}}$	−103.09 mm	−148.21 mm
$iFEM^{P3_{\max}}$	−104.26 mm	−149.03 mm
$RRMSE^{P1}$	0.037	0.039
$RRMSE^{P3}$	0.042	0.047

It can be observed from Table 6 that under the load of 200 N, the maximum experimental deformation NDI^{\max} is 100.29 mm. With scheme P_1 , the maximum reconstructed displacement $iFEM^{\max}$ is 103.09 mm, while with scheme P_3 , $iFEM^{\max}$ is 104.26 mm. The error is controlled within 4 mm. Under the load of 350 N, the NDI^{\max} is 144.09 mm, the $iFEM^{\max}$ with scheme P_1 is 148.21 mm, while $iFEM^{\max}$ with scheme P_3 is 149.21 mm, the error is controlled within 5 mm, and the $RRMSE$ can also be kept within 5%. The results demonstrate that the reconstruction effect of the sensor schemes P_1 and P_3 is excellent.

Since the reconstruction accuracy of the sensor layout scheme P_1 is similar to that of the sensor layout scheme P_3 , the scheme P_1 is selected for visualization. Figure 14 shows the deformation measured by NDI and reconstructed deformation with scheme P_1 under two concentrated loading conditions, and the reconstructed deformation is coincided with the reference value from the NDI, which indicates that the reconstruction effect is excellent with scheme P_1 .

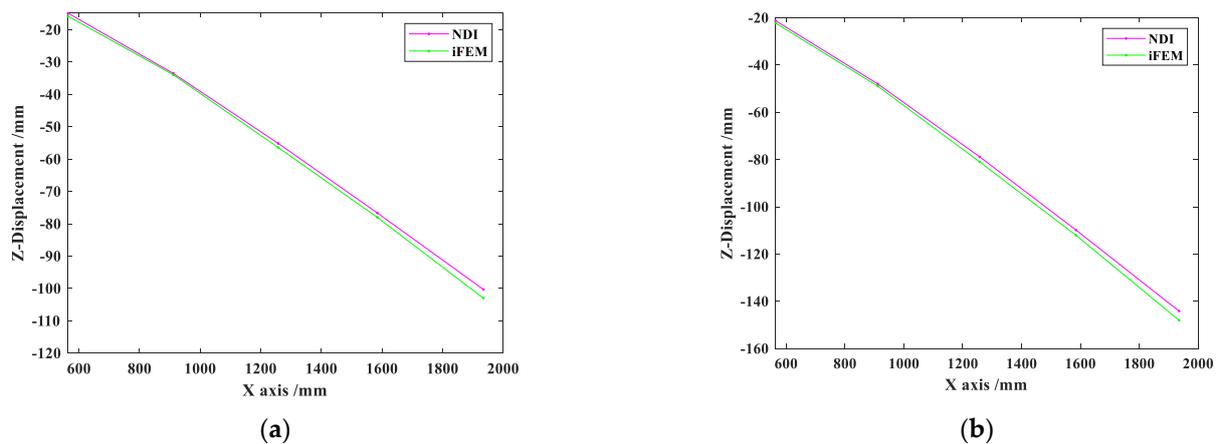


Figure 14. Displacement in Z direction with P_1 under concentrated load of 200 N (a) and 350 N (b).

In conclusion, IAMOPSO can well solve the sensor distribution problem in iFEM. In practical engineering applications, under complex working conditions, on the basis of meeting the accuracy requirements, a sensor distribution scheme with higher stability should be selected, while under other conditions, a sensor distribution with high accuracy has priority.

5. Conclusions

This paper focuses on an optimization algorithm that can be better applied to sensor layout problems. This paper first constructs a multi-objective optimization model of sensor distribution in iFEM based on the accuracy and robustness metrics. To balance the optimization of the two objectives and improve the efficiency of the algorithm, this paper proposes an improved adaptive multi-objective particle swarm optimization algorithm, which introduces an initialization strategy, adaptive inertia weight strategy, guided particle selection strategy, and ECS set maintenance strategy. The results on the beam model validate the performance of the algorithm. Finally, the algorithm is applied to the deformation reconstruction of the complex plate-beam model, which can maintain high reconstruction accuracy and stability. The numerical and experimental results demonstrate that the optimization algorithm IAMOPSO proposed in this paper can well solve the sensor distribution problem in iFEM. Future work will focus on solving high-dimensional optimization problems in complex structural reconstruction.

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

Funding: This research was funded by National Natural Science Foundation of China, grant number 51775401.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Roshan, J.; Mei, H.F.; Asaad, M.; Victor, G. Crack-Length Estimation for Structural Health Monitoring Using the High-Frequency Resonances Excited by the Energy Release during Fatigue-Crack Growth. *Sensors* **2021**, *21*, 4221.
2. Das, S.; Saha, P. Performance of swarm intelligence based chaotic meta-heuristic algorithms in civil structural health monitoring. *Measurement* **2021**, *169*, 108533. [[CrossRef](#)]
3. Gerardo, A.; Javier, B.; Josu, E.; Miguel, G.J.; Carolina, R. Methodology for Detecting Progressive Damage in Structures Using Ultrasound-Guided Waves. *Sensors* **2022**, *22*, 1692.
4. Guo, J.; Zhang, L.L.; Jiang, H. Research on Real-time Signal Stability of Computer Monitoring Bridge Structure Health Sensor. *Sci. J. Intell. Syst. Res.* **2022**, *4*, 256–261.
5. Huang, H.B.; Yi, T.H.; Li, H.N. Anomaly Identification of Structural Health Monitoring Data Using Dynamic Independent Component Analysis. *J. Comput. Civ. Eng.* **2020**, *34*, 04020025. [[CrossRef](#)]
6. Azimi, M.; Pekcan, G. Structural health monitoring using extremely compressed data through deep learning. *Comput. Aided Civ. Infrastruct. Eng.* **2020**, *35*, 597–614. [[CrossRef](#)]
7. Guo, C.; Zou, J.P.; Sun, K. Analysis on Structural Health Monitoring System of High-Pile Wharf Based on Optical Fiber Sensor. *J. Phys. Conf. Ser.* **2021**, *1881*, 042018. [[CrossRef](#)]
8. Gherlone, M.; Cerracchio, P.; Mattone, M. Shape sensing methods: Review and experimental comparison on a wing-shaped plate. *Prog. Aerosp. Sci.* **2018**, *99*, 14–26.
9. Tessler, A.; Spangler, J.L. *A Variational Principle for Reconstruction of Elastic Deformations in Shear Deformable Plates and Shells*; NASA/TM-2003-212445; National Aeronautics and Space Administration, Langley Research Center: Hampton, VA, USA, 2003.
10. Tessler, A.; Spangler, J.L. A least-squares variational method for full-field reconstruction of elastic deformations in shear-deformable plates and shells. *Comput. Methods Appl. Mech. Eng.* **2005**, *194*, 327–339. [[CrossRef](#)]
11. Gherlone, M.; Cerracchio, P.; Mattone, M.; Sciuva, M.D.; Tessler, A. Shape sensing of 3D frame structures using an inverse Finite Element Method. *Int. J. Solids Struct.* **2012**, *49*, 3100–3112. [[CrossRef](#)]
12. Gherlone, M.; Cerracchio, P.; Mattone, M.; Di Sciuva, M.; Tessler, A. An inverse finite element method for beam shape sensing: Theoretical framework and experimental validation. *Smart Mater. Struct.* **2014**, *23*, 045027. [[CrossRef](#)]
13. Zhao, Y.; Bao, H.; Duan, X.C.; Fang, H.M. The Application Research of Inverse Finite Element Method for Frame Deformation Estimation. *Int. J. Aerosp. Eng.* **2017**, *2017*, 1–8. [[CrossRef](#)]
14. Niu, S.T.; Zhao, Y.; Bao, H. Shape sensing of plate structures through coupling inverse finite element method and scaled boundary element analysis. *Measurement* **2022**, *190*, 110676. [[CrossRef](#)]
15. Chen, K.Y.; He, D.H.; Zhao, Y.; Bao, H.; Du, J.L. A Unified Full-Field Deformation Measurement Method for Beam-Like Structure. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–10. [[CrossRef](#)]
16. Yuen, K.V.; Kuok, S.C. Efficient Bayesian sensor placement algorithm for structural identification: A general approach for multi-type sensory systems. *Earthq. Eng. Struct. Dyn.* **2015**, *44*, 757–774. [[CrossRef](#)]
17. Zhao, Y.; Du, J.L.; Bao, H.; Xu, Q.; Christopher, J.D. Optimal Sensor Placement for Inverse Finite Element Reconstruction of Three-Dimensional Frame Deformation. *Int. J. Aerosp. Eng.* **2018**, *2018*, 6121293. [[CrossRef](#)]
18. Zhao, F.F.; Bao, H. Multi-Objective Particle Swarm Optimization of Sensor Distribution Scheme with Consideration of the Accuracy and the Robustness for Deformation Reconstruction. *Sensors* **2019**, *19*, 1306. [[CrossRef](#)]
19. Cheshmehgaz, H.R.; Haron, H.; Sharifi, A. The review of multiple evolutionary searches and multi-objective evolutionary algorithms. *Artif. Intell. Rev.* **2015**, *43*, 311–343. [[CrossRef](#)]
20. Chen, Y.K.; Huang, S.; Davis, L.; Du, H.; Shi, Q.; He, J.; Wang, Q.; Hu, W. Optimization of Geometric Parameters of Longitudinal-Connected Air Suspension Based on a Double-Loop Multi-Objective Particle Swarm Optimization Algorithm. *Appl. Sci.* **2018**, *8*, 1454. [[CrossRef](#)]
21. Akbarzadeh, P.; Mikaeeli, S.Z.; Rahimiyan, M. Multiobjective optimization of thermohydrodynamic journal bearing using MOPSO algorithm. *Proc. Inst. Mech. Eng. Part J J. Eng. Tribol.* **2018**, *232*, 657–671. [[CrossRef](#)]
22. Sreekara Reddy, M.B.S.; Ratnam, C.; Rajyalakshmi, G.; Manupati, V.K. An effective hybrid multi objective evolutionary algorithm for solving real time event in flexible job shop scheduling problem. *Measurement* **2018**, *114*, 78–90. [[CrossRef](#)]
23. Liu, J.F.; Liu, J. Applying multi-objective ant colony optimization algorithm for solving the unequal area facility layout problems. *Appl. Soft Comput. J.* **2018**, *74*, 167–189. [[CrossRef](#)]
24. Lin, C.H.; Hwang, C.C. High Performances Design of a Six-Phase Synchronous Reluctance Motor Using Multi-Objective Optimization with Altered Bee Colony Optimization and Taguchi Method. *Energies* **2018**, *11*, 2716. [[CrossRef](#)]
25. Zhou, J.Z.; Lu, P.; Li, Y.Z.; Wang, C.; Yuan, L.; Mo, L. Short-term hydro-thermal-wind complementary scheduling considering uncertainty of wind power using an enhanced multi-objective bee colony optimization algorithm. *Energy Convers. Manag.* **2016**, *123*, 116–129. [[CrossRef](#)]
26. Han, F.; Chen, W.T.; Ling, Q.H.; Han, H. Multi-objective particle swarm optimization with adaptive strategies for feature selection. *Swarm Evol. Comput.* **2021**, *62*, 100847. [[CrossRef](#)]
27. Mandal, P.; Mondal, S.C. Multi-objective optimization of Cu-MWCNT composite electrode in electro discharge machining using MOPSO-TOPSIS. *Measurement* **2021**, *169*, 108347. [[CrossRef](#)]
28. Peng, S.; Xiong, Y. An Area Coverage and Energy Consumption Optimization Approach Based on Improved Adaptive Particle Swarm Optimization for Directional Sensor Networks. *Sensors* **2019**, *19*, 1192. [[CrossRef](#)]

29. Luu, K.; Noble, M.; Gesret, A.; Belayouni, N.; Roux, P.F. A parallel competitive Particle Swarm Optimization for non-linear first arrival travelttime tomography and uncertainty quantification. *Comput. Geosci.* **2018**, *113*, 81–93. [[CrossRef](#)]
30. Lee, J.; Park, J.; Kim, H.C.; Kim, D.W. Competitive Particle Swarm Optimization for Multi-Category Text Feature Selection. *Entropy* **2019**, *21*, 602. [[CrossRef](#)]
31. Deng, L.B.; Song, L.; Sun, G.J. A Competitive Particle Swarm Algorithm Based on Vector Angles for Multi-Objective Optimization. *IEEE Access* **2021**, *9*, 89741–89756. [[CrossRef](#)]
32. Yu, X.; Wang, H.; Sun, H. Decomposition-based multi-objective comprehensive learning particle swarm optimisation. *Int. J. Comput. Sci. Eng.* **2019**, *18*, 349–360. [[CrossRef](#)]
33. Zhu, Q.L.; Lin, Q.Z.; Chen, W.N.; Wong, K.C.; Coello, C.C.A.; Li, J.Q.; Chen, J.Y.; Zhang, J. An External Archive-Guided Multiobjective Particle Swarm Optimization Algorithm. *IEEE Trans. Cybern.* **2017**, *47*, 2794–2808. [[CrossRef](#)]
34. Li, L.; Wang, W.L.; Li, W.K.; Xu, X.L.; Zhao, Y.W. A Novel Ranking-based Optimal Guides Selection Strategy in MOPSO. *Procedia Comput. Sci.* **2016**, *91*, 1001–1010. [[CrossRef](#)]
35. Gordis, J.H.; Jae-Cheol, S.; Bouwense, M.D. Minimum condition number by orthogonal projection row selection of artificial boundary conditions for finite element model update and damage detection. *J. Sound Vib.* **2018**, *433*, 179–197. [[CrossRef](#)]
36. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the Icn95-International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
37. Engelbrecht, A.P. Particle swarm optimization with crossover: A review and empirical analysis. *Artif. Intell. Rev.* **2016**, *45*, 131–165. [[CrossRef](#)]