

## Article

# A Large-Scale Sensor Layout Optimization Algorithm for Improving the Accuracy of Inverse Finite Element Method

Zhenyi Zhao <sup>1</sup>, Kangyu Chen <sup>1</sup>, Yimin Liu <sup>1</sup> and Hong Bao <sup>1,2,\*</sup>

<sup>1</sup> Key Laboratory of Electronic Equipment Structure Design, Ministry of Education, Xidian University, Xi'an 710071, China; 22041212766@stu.xidian.edu.cn (Z.Z.); ky.chen@stu.xidian.edu.cn (K.C.); 22041212778@stu.xidian.edu.cn (Y.L.)

<sup>2</sup> Intelligent Robot Laboratory, Hangzhou Research Institute of Xidian University, Hangzhou 311231, China

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

**Abstract:** The inverse finite element method (iFEM) based on fiber grating sensors has been demonstrated as a shape sensing method for health monitoring of large and complex engineering structures. However, the existing optimization algorithms cause the local optima and low computational efficiency for high-dimensional strain sensor layout optimization problems of complex antenna truss models. This paper proposes the improved adaptive large-scale cooperative coevolution (IALSCC) algorithm to obtain the strain sensors deployment on iFEM, and the method includes the initialization strategy, adaptive region partitioning strategy, and *gbest* selection and particle updating strategies, enhancing the reconstruction accuracy of iFEM for antenna truss structure and algorithm efficiency. The strain sensors optimization deployment on the antenna truss model for different postures is achieved, and the numerical results show that the optimization algorithm IALSCC proposed in this paper can well handle the high-dimensional sensor layout optimization problem.

**Keywords:** inverse finite element method; cooperative coevolution; particle swarm optimization; grouping method; structural health monitoring



**Citation:** Zhao, Z.; Chen, K.; Liu, Y.; Bao, H. A Large-Scale Sensor Layout Optimization Algorithm for Improving the Accuracy of Inverse Finite Element Method. *Sensors* **2023**, *23*, 8176. <https://doi.org/10.3390/s23198176>

Academic Editors: Jose Alfonso Antonino-Daviu, Francesc Pozo, Steven Chatterton and Mohammad N Noori

Received: 2 September 2023

Revised: 21 September 2023

Accepted: 27 September 2023

Published: 29 September 2023



**Copyright:** © 2023 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

In recent decades, the application of structural health monitoring (SHM) has gained considerable attention in various fields such as large-scale bridges, high-rise buildings, and aerospace. The practical implementation techniques have also been investigated by engineering researchers [1–4]. SHM provides a strong technical guarantee for structural safety through real-time monitoring and analysis of various structural data, timely detection of structural defects and hidden dangers, and prediction of structural life and fatigue state. In Gherlone's research on deformation reconstruction methods [5], shape sensing was utilized to reconstruct the deformation field of a structure by measuring real-time strain values using strain sensors attached to different positions of the structure.

The key aspect of shape sensing is to establish an accurate mathematical relationship between the measured surface strain values and the reconstructed displacement field. Various modeling approaches have been proposed by researchers for this purpose. Some authors have employed methods such as fuzzy networks and neural networks for modeling [6,7]. These approaches require a substantial amount of training samples and the quality of these samples affects the accuracy of the reconstruction. In the literature [8,9], the surface-measured strain is fitted to obtain the structural strain field using global or segmented continuous basis function methods. Subsequently, the displacement field of the structure is obtained through the strain–displacement relationship. This approach is easily implementable, but the accuracy of the reconstruction relies on the selection of basic functions and weight coefficients. Jineesh et al. selected modal shapes as basic functions and reconstructed the displacement field using modal transformation techniques [10,11].

This method exhibits better reconstruction accuracy due to its inherent modal characteristics. However, the drawback lies in the requirement of detailed material properties for constructing modal shapes, making it less applicable to complex structures.

Tessler and Spangler proposed the inverse finite element method (iFEM) based on the first-order shear deformation theory [12,13]. This method offers the advantage of accomplishing deformation reconstruction without requiring prior knowledge of load, material parameters, and structural properties. The framework of the iFEM was extended by Gherlone for the beam element, who validated the feasibility of their approach by conducting shape sensing analysis on a cantilever beam structure subjected to static and dynamic loads [14]. In [15], shape sensing analysis was conducted on a typical composite reinforced structure using the iFEM, demonstrating the superiority of the iFEM. Data collection from individual surfaces of the structure was employed by Niu et al. to reconstruct the displacement field [16], thus reducing the number of sensors used. Chen et al. introduced a unified displacement field for beam-like structures based on the homogenization theory, utilizing certain generalized quantities instead of traditional displacement functions to avoid reconstruction errors caused by structural identification inaccuracies [17]. Since the measured strain values from strain sensors serve as crucial data for shape sensing techniques, the layout positions of strain sensors have a significant impact on the reconstruction accuracy. Feng et al. proposed a method based on numerical optimization to obtain the best sensor position [18]. Zhao et al. optimized the layout of sensors using a single-objective particle swarm optimization (SOPSO) algorithm, targeting well-separated eigenvalues [19]. However, using a SOPSO algorithm makes it challenging to strike a balance between robustness and accuracy. In [20], the optimization was performed using a multi-objective particle swarm algorithm (MOPSO) with robustness and accuracy as optimization objectives, but the absence of an appropriate strategy to ensure diversity resulted in a tendency to get trapped in the local optima. By introducing strategies such as guided particle selection and maintenance of an external candidate solution set, Li et al. improved the optimization performance [21]. The above algorithms improve the optimization by introducing different methods, but the effectiveness of these methods significantly diminishes when the dimensionality of the solution space becomes too large.

Optimizing sensor layout with reconstruction accuracy and robustness as optimization objectives constitutes a multi-objective optimization problem (MOP). Currently, the adoption of cooperative coevolution (CC) is considered as a successful approach for addressing large-scale multi-objective optimization problems (LSMOPs). The CC method was proposed by Potter and De Jong in [22], where the idea is to decompose the decision variables using various variable grouping methods and optimize them independently. The CC framework is primarily composed of three components: grouping strategy, optimizer, and collaboration method. The grouping strategy is a method for decomposing a large number of decision variables, and existing grouping strategies include fixed grouping; random grouping [23]; linear grouping [24]; ordered method [25]; and dynamic grouping [26]. These groupings are usually simpler, but since they do not take into account the attributes of the decision variables, their enhancement effect on optimization is small. The optimizer is the algorithm used to optimize each subgroup, such as the aforementioned MOPSO algorithm [20,21]; they perform poorly when faced with high-dimensional optimization problems. The collaboration method defines the inter-group collaboration strategy, which specifies the content and manner of information sharing between subgroups [27]. The objective of this paper is to propose an optimization algorithm that can be applied to large-scale sensor layout optimization problems.

This paper improves and optimizes the above two parts respectively. In terms of grouping strategy, citing the concept of the convergence relevance degree (CRD) proposed by Ma et al. in [28], on the basis of calculating the CRD value of each decision variable, a grouping method according to the size of the CRD value is proposed; this method can make full use of the attribute characteristics of decision variables and attribute decision variables with similar characteristics to the same subgroup. In terms of the optimizer,

in order to balance the diversity and convergence of the population and prevent the optimization result from falling into a local optimum, a method of adaptive region division based on the number of external archive particles is designed, and particle updates in each region are guided by the globally optimal particles in other regions. The advantage of this method is that it can fully explore the target space during the optimization process and improve the search efficiency by limiting the number of non-dominated solutions in the external archive.

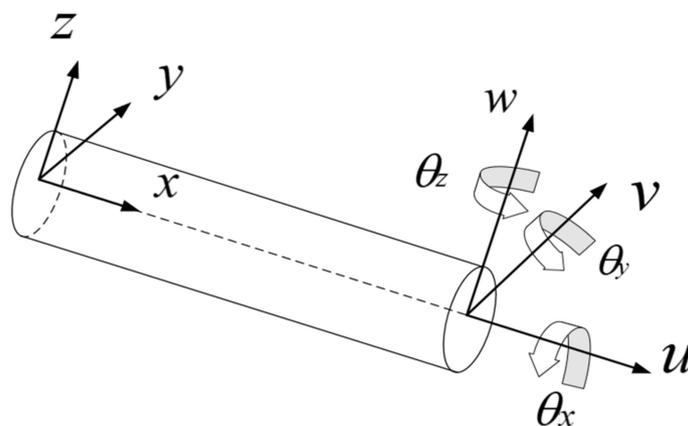
The organization of this paper is as follows. In Section 2, the iFEM for the basic beam element is introduced, and a sensor layout optimization model based on reconstruction accuracy and robustness is established. In Section 3, the improved adaptive large-scale cooperative coevolution (IALSCC) algorithm is described in detail. In Section 4, IALSCC is applied to the sensor layout problem for deformation reconstruction of complex antenna truss structure, and simulation data verifies that IALSCC is a promising tool for sensor layout optimization.

## 2. Optimization Model Based on iFEM for Beam Structures

### 2.1. Inverse Finite Element Method

According to the Timoshenko beam theory, as shown in Figure 1, the deformation of any point on the surface of the beam can be represented by displacements along the respective axes and rotations about the respective axes. These six kinematic variables can be grouped in vector form:

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



**Figure 1.** Schematic diagram of Timoshenko beam.

Based on the kinematic assumption of three-dimensional deformation, the displacement vector at any point on the beam cross-section can be represented by the displacement of the neutral axis:

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

Based on the assumption of small strains, the strain vector at any cross-section is defined as

$$\mathbf{e}(\mathbf{u}) = [e_1, e_2, e_3, e_4, e_5, e_6]^T, \quad (3)$$

where  $e_1$  is the cross-sectional strain caused by the elongation deformation of the element,  $e_2$  and  $e_3$  are the cross-sectional strains caused by the bending deformation of the element,

$e_4$  and  $e_5$  are related to the shear deformation, and  $e_6$  is related to the torsional deformation of the element. The cross-sectional strains can be obtained from Equation (2):

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

The strain along the axis at any point is obtained by taking the derivative of Equation (2) with respect to  $x$ :

$$\begin{cases} \varepsilon_x(x, y, z) = e_1(x) + ze_2(x) + ye_3(x) \\ \gamma_{xz}(x, y) = e_4(x) + ye_6(x) \\ \gamma_{xy}(x, y) = e_5(x) - ze_6(x) \end{cases} \quad (5)$$

The basic framework of the iFEM is to minimize the error between experimental and theoretical strains using the least squares method. The fitting function  $\varphi(\mathbf{u})$  is defined by the theoretical sectional strain  $\mathbf{e}(\mathbf{u})$  and the actual sectional strain  $\mathbf{e}^\varepsilon$ :

$$\varphi(\mathbf{u}) = \|\mathbf{e}(\mathbf{u}) - \mathbf{e}^\varepsilon\|^2. \quad (6)$$

By taking the derivative of Equation (6) with respect to  $\mathbf{u}$  and setting the derivative equal to 0, the relationship between the neutral axis displacement variable of the beam and the experimental strain measurements is established:

$$\mathbf{k}^e \mathbf{u}_e = \mathbf{f}^e. \quad (7)$$

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

$$\begin{cases} \mathbf{k}^e = \sum_{k=1}^6 w_k \mathbf{k}_k^e, \mathbf{k}_k^e = \sum_{i=1}^n \frac{L}{n} [\mathbf{B}_k^T(x_i) \mathbf{B}_k(x_i)] \\ \mathbf{f}^e = \sum_{k=1}^6 w_k \mathbf{f}_k^e, \mathbf{f}_k^e = \frac{L}{n} \sum_{i=1}^n [\mathbf{B}_k^T(x_i) \mathbf{e}_k^\varepsilon(x_i)] \end{cases} \quad (8)$$

where  $n$  represents the number of sections,  $L$  is the length of the element,  $x_i$  is the calculation position of the strain,  $\mathbf{e}_k^\varepsilon(x_i)$  is the calculated strain at position  $x_i$  obtained from the measured strain data,  $w_k (k = 1, 2, \dots, 6)$  are the weighting coefficients that consider the mutual influence of axial stretching, bending, torsion, and transverse shear,  $\mathbf{B}_k(x_i)$  is the coefficient matrix obtained by taking the derivative of the shape function matrix, and once  $x_i$  is determined, the coefficient matrix is also determined. It can be observed that  $\mathbf{k}^e$  is a function of  $x_i$ , and  $\mathbf{f}^e$  is determined by  $x_i$  and the measured strain value  $\varepsilon_2^*$ .

Equation (9) is obtained by assembling multiple inverse finite element units:

$$\mathbf{KU} = \mathbf{F}, \quad (9)$$

where  $\mathbf{K}$  is the overall class stiffness matrix, determined by the shape functions and the positions of strain sensors, and  $\mathbf{F}$  is the global class load vector, which depends only on the measured strain. This equation provides the solution for the degrees of freedom of the structural deformation nodes,  $\mathbf{U} = \mathbf{K}^{-1}\mathbf{F}$ . Then, since the displacement at any cross-section location can be expressed using shape functions and nodal degrees of freedom, the deformation at any cross-section can be obtained using interpolation methods. Finally, based on Equation (2), the deformations at arbitrary points can be calculated.

## 2.2. Metrics to Evaluate Refactoring Effects

After reconstructing the displacement field of the structure, it is necessary to evaluate the performance of the reconstructed field. In this study, the root mean square error (RMSE) and robustness index (RBI) are employed to assess the accuracy of the reconstruction and the stability of the reconstruction model, respectively.

Reconstruction accuracy is a key indicator for evaluating the effectiveness of the iFEM, is measured by the *RMSE*, and defined as follows:

$$RMSE(\mathbf{X}) = \sqrt{\frac{1}{N} \sum_{i=1}^N (disp_i^{measurement}(\mathbf{X}) - disp_i^{iFEM}(\mathbf{X}))^2}, \quad (10)$$

where  $\mathbf{X}$  represents the sensor layout scheme,  $N$  is the number of reference points,  $disp^{measurement}$  is the actual measured value obtained through simulation software or measurement devices,  $disp^{iFEM}$  is the predicted value derived using the iFEM, and  $disp(\mathbf{X})$  is the deformation displacement along the centroid axis. The key step of iFEM is to calculate the cross-section strain based on the measured surface strain data, and the placement of the strain sensor directly affects the value of the surface strain and the calculation of the class stiffness matrix, which in turn affects the reconstruction accuracy of the displacement field.

Robustness is also an important indicator for evaluating the effectiveness of deformation reconstruction, which reflects the ability of the model to maintain performance stability in the face of uncertainty and perturbations. In order to clearly represent this ability and optimize it as an objective, a method for quantifying reconstruction stability introduced in [29] is employed.

$$\cos(\theta_i(\mathbf{X})) = \left| \frac{\mathbf{A}_i(\mathbf{X})\mathbf{P}(\mathbf{X})\mathbf{A}_i^T(\mathbf{X})}{\|\mathbf{A}_i(\mathbf{X})\| \times \|\mathbf{P}(\mathbf{X})\mathbf{A}_i^T(\mathbf{X})\|} \right|, \quad (11)$$

$$\mathbf{P}_i(\mathbf{X}) = \bar{\mathbf{A}}_i^T(\mathbf{X}) \left( \bar{\mathbf{A}}_i(\mathbf{X})\bar{\mathbf{A}}_i^T(\mathbf{X}) \right)^{-1} \bar{\mathbf{A}}_i(\mathbf{X}), \quad (12)$$

where the matrix  $\mathbf{A}(\mathbf{X})$  is a class stiffness matrix that is only dependent on the sensor layout.  $\mathbf{A}_i(\mathbf{X})$  represents the  $i$ -th row vector of matrix  $\mathbf{A}(\mathbf{X})$ , while  $\bar{\mathbf{A}}_i(\mathbf{X})$  is a matrix composed of the remaining row vectors excluding  $\mathbf{A}_i(\mathbf{X})$ . The larger the angle  $\theta_i(\mathbf{X})$  becomes, the lower the correlation between the vector matrices, indicating a stronger ability to withstand external perturbations. By calculating the angle  $\theta_i(\mathbf{X})$  for each row vector, the *RBI* is defined as the minimum value among them:

$$RBI(\mathbf{X}) = \min(\theta_i(\mathbf{X})) \quad i = 1, 2, \dots, m, \quad (13)$$

where  $m$  represents the dimension of the node degrees of freedom. The larger the *RBI*( $\mathbf{X}$ ) value, the stronger the anti-interference ability of this sensor layout scheme.

After defining the two objective functions, the sensor layout scheme that balances reconstruction accuracy and robustness can be sought through an optimization algorithm. The optimization model is defined as follows:

$$\mathbf{X} = \operatorname{argmin} \{RMSE(\mathbf{X}), -RBI(\mathbf{X})\}, \quad (14)$$

where  $\mathbf{X}$  represents the layout positions of the sensors with dimensions equivalent to the number of sensors. In the third section, the proposed IALSCC algorithm will be thoroughly described.

### 3. Improved Adaptive Large-Scale Cooperative Coevolutionary Algorithm

The algorithm proposed in this paper focuses on optimizing and improving two components: the optimizer and the grouping method. The optimizer is a MOPSO algorithm, while the grouping method addresses the issue of grouping a large number of decision variables. In Section 3.1, the MOPSO algorithm will be introduced. In Section 3.2, the improved multi-objective particle swarm optimization algorithm proposed in this paper will be discussed. Section 3.3 will present the grouping method based on CRD calculation and difference degree (DD). Finally, Section 3.4 outlines the algorithm framework of IALSCC.

### 3.1. Introduction of MOPSO

PSO is an intelligent algorithm inspired by the foraging behavior of birds, originally proposed by Kennedy and Eberhart in [30]. The principle is that each particle is guided to update itself based on its own historical information and the information of the swarm. The expression of the algorithm is given by the following equations:

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 r_1 (pbest_{id}^t - x_{id}^t) + c_2 r_2 (gbest_d^t - x_{id}^t), \quad (15)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}, \quad (16)$$

where  $v_{id}^t$  is the  $d$ -th component of the velocity vector of particle  $i$  at the  $t$ -th iteration;  $x_{id}^t$  is the  $d$ -th component of the position vector of particle  $i$  at the  $t$ -th iteration;  $pbest_{id}^t$  is the  $d$ -th component of the optimal position vector of particle  $i$  at the  $t$ -th iteration;  $gbest_d^t$  is the  $d$ -th component of the optimal position vector of the population at the  $t$ -th iteration;  $c_1$  and  $c_2$  are self-awareness coefficient and group learning coefficient, respectively;  $r_1$  and  $r_2$  are two random numbers, the value range is  $[0, 1]$ ; and  $\omega$  is a inertia weight, which adjusts the search range of the solution space.

The MOPSO algorithm is an optimization algorithm proposed based on the PSO algorithm to address MOPs [31]. A MOP can be described as a minimization problem, defined as follows:

$$\begin{aligned} \text{Minimize } F(\mathbf{X}) &= \{F_1(\mathbf{X}), F_2(\mathbf{X}), \dots, F_n(\mathbf{X})\} \\ \text{subject to } \mathbf{X} &\in \Omega \end{aligned} \quad (17)$$

where  $n$  is the dimension of the objective space and  $\Omega$  is the decision variable space. Obviously, the existence of multiple objective functions makes it difficult to compare the quality of solutions. The Pareto dominance relation is a commonly used method for comparing the quality of solutions, which is defined as if  $X, Y \in \Omega$ ,  $\forall k \in \{1, 2, \dots, m\}$ ,  $F_k(X) \leq F_k(Y)$  and  $\exists k \in \{1, 2, \dots, m\}$ ,  $F_k(X) < F_k(Y)$ ;  $m$  is the number of decision variables and  $Y$  is said to be dominated by  $X$ , otherwise there is a non-dominated relationship between  $X$  and  $Y$ .

The balance between diversity and convergence is a crucial aspect in ensuring the effectiveness of the MOPSO. If emphasis is placed on convergence while neglecting exploration of the objective space, the results may easily become trapped in local optima. Conversely, if focus is solely on exploration without sufficient convergence, the exploration of the objective space may prove ineffective. The proposed improved algorithm in this paper effectively balances these two aspects and demonstrates excellent performance in handling MOPs.

### 3.2. Strategies Used by the Proposed Improved MOPSO

#### 3.2.1. Initialization Strategy

In traditional MOPSO algorithms, the velocity and position of particles are typically randomly initialized. In this algorithm, the initial region partitioning in the objective space is based on the distribution of particles. Uneven distribution of particles can result in a low number of particles entering the optimization process, leading to decreased exploration efficiency. Therefore, it is necessary to improve the initialization strategy to achieve a more uniform distribution of initial particles in the objective space.

Using boundary particles as global best particles to guide the population for early-stage updates is an effective method. Specifically, assuming there are  $n$  optimization objectives, during the initialization stage,  $n$  rounds of single-objective optimization are performed. It is important to note that after completing each round of single-objective optimization, the results of each iteration in that round are saved. Then, the next single-objective optimization should begin from the initial state rather than from the previous optimization results. The final set of particles is the union of the saved results. Additionally, the number of iterations  $T_{ini}$  and the initial population size  $N_{ini}$  need to be appropriately

chosen. If the parameters are not suitable and result in a large population size  $N$ , it can affect the efficiency of the algorithm. Conversely, if  $N$  is too small, the uniformization effect may not be satisfactory. Generally,  $N_{ini} < N/5$  and  $T_{ini} < T/10$  are considered appropriate, where  $T$  is the maximum number of iterations.

By employing this initialization strategy, the distribution of particles in the objective space can be made more uniform, thus enhancing the population diversity and the performance of the optimizer in MOPSO.

### 3.2.2. Adaptive Region Partitioning Strategy

In this paper, a method is proposed for adaptive region partitioning based on the distribution of particles in an external archive set. Taking bi-objective optimization as an example, after uniform initialization of particles, non-dominated solutions are selected based on the Pareto dominance relation and stored in the external archive set.

The external archive set is divided into  $D_i$  regions based on the given number of regions, and it is important to note that normalization of the objective space is performed during the region partitioning to prevent some particles from being neglected in the calculation.

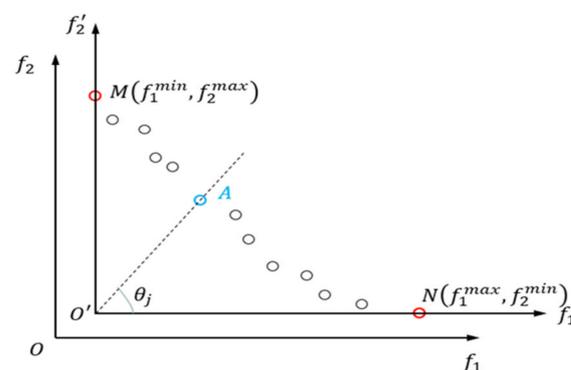
As shown in Figure 2, when optimizing the objective  $f_1$ , point  $M$  has the minimum fitness value  $f_1^{min}$ . When optimizing the objective  $f_2$ , point  $N$  has the minimum fitness value  $f_2^{min}$ . After normalization, the coordinate origin shifts from the original point  $O$  to  $O'(f_1^{min}, f_2^{min})$ . Then, in the normalized coordinate system, the angle  $\theta_j$  between the line connecting the non-dominated particle and the origin and the horizontal axis is calculated using Equation (18):

$$\theta_j = \arctan\left(\frac{f_2(x_j) - f_2^{min}}{f_1(x_j) - f_1^{min}}\right), \quad (18)$$

where  $f_2(x_j)$  represents the fitness of the  $j$ -th particle in optimization objective  $f_2$  and  $f_1(x_j)$  represents the fitness of the  $j$ -th particle in optimization objective  $f_1$ . Finally, based on Equation (19), it is determined whether each region contains particles. For regions that contain particles, one particle is selected from each region, and their union forms the population for formal optimization. These particles represent high-quality information from the initial population, and utilizing them can improve the efficiency of the algorithm. Equation (19) is as follows:

$$0.5 \times \pi \times (i - 1)/D_i \leq \theta_j < 0.5 \times \pi \times i/D_i, \quad (19)$$

where  $i$  represents the  $i$ -th region and  $D_i$  represents the total number of divided regions.



**Figure 2.** Normalized coordinate establishment.

After the aforementioned steps, we can calculate that among the  $D_i$  regions in the external archive set there are  $space\_n$  regions that already contain non-dominated solutions. Then, based on the range of  $space\_n$ , the objective space is divided into  $D_j \in [1, 4(2^n)]$  regions. The specific partitioning method is as follows.

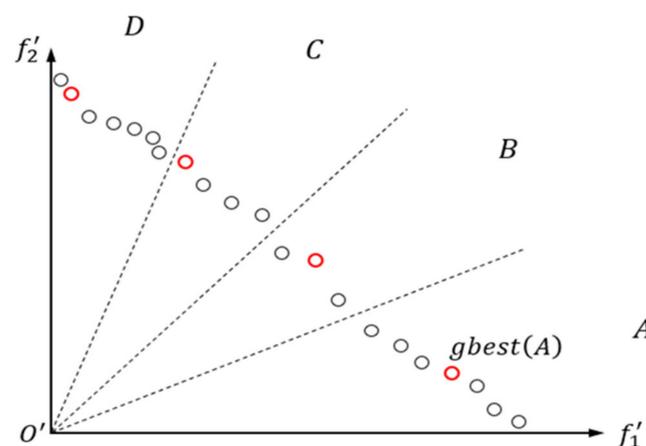
If  $0 < space\_n \leq 1/4 \times D_i$ , then  $D_j = 4$ . In this case, the value of  $space\_n$  is small, indicating that non-dominated particles are not present in most regions. There are two possibilities: one is that non-dominated solutions do not exist in that region, indicating that the Pareto Front (PF) is discontinuous; the other is that the region has not been explored, suggesting insufficient population diversity, which requires enhancing exploration in that region to improve diversity. If  $1/4 \times D_i < space\_n \leq 2/4 \times D_i$ , then  $D_j = 3$ ; if  $2/4 \times D_i < space\_n \leq 3/4 \times D_i$ , then  $D_j = 2$ ; if  $3/4 \times D_i < space\_n \leq 1 \times D_i$  and  $D_j = 1$ . In the latter case, where most regions already meet the condition of having non-dominated solutions, there is no need for further processing of the objective space.

After the initial region partitioning, it is necessary to find  $gbest$  to guide particle updates. With each iteration process completed, the value of  $space\_n$  is recalculated and the objective space is adaptively partitioned based on the value of  $space\_n$ . The selection strategy for  $gbest$  will be described in detail in the next section.

### 3.2.3. Selection of $gbest$ and Particle Update Strategies

The selection of  $gbest$  has a significant impact on the optimization results. In previous algorithms, typically only one  $gbest$  is selected in the objective space to guide the optimization process, aiming to ensure convergence. However, in high-dimensional optimization problems, due to the large number of decision variables, the solution space becomes rapidly complicated and it is more likely that the optimization result will fall into the local optimum. Therefore, on the basis of considering convergence, this method places more emphasis on improving population diversity.

In Section 3.2.2, the objective space has been divided into  $D_j$  regions, taking  $D_j = 4$  as an example. A non-dominated solution is randomly selected from each of the four regions as the  $gbest$  for that region. Then, the  $gbest$  of each region is used to guide the particle updates in other regions. The purpose of this operation is to allow particles to thoroughly explore the objective space and obtain potentially more promising non-dominated solutions. As shown in Figure 3,  $gbest(A)$  is randomly selected in region A, and it is used to guide the particle updates in regions other than A (e.g., region D). It should be noted that if there are no particles in a particular region, resulting in the absence of  $gbest$  for that region, the  $gbest$  from the adjacent region is used as a substitute. The advantage of this approach is its ability to thoroughly explore the objective space, significantly enhancing the diversity of the population.



**Figure 3.**  $Gbest$  selection and particle update strategies ( $D_j = 4$ ).

After each iteration, the non-dominated particles are saved to the external archive. Then, based on the Pareto dominance relationship, all non-dominated particles in the external archive are filtered out. The external archive contains all the non-dominated particles found so far in the search. It is important to note that as the iterations progress, the number of non-dominated solutions in the external archive grows rapidly. Therefore, it is necessary

to set a threshold to prevent a decrease in algorithm efficiency. The threshold needs to be properly set, as a too-low threshold may result in a high deletion frequency of particles, potentially leading to the removal of some high-quality particles. On the other hand, a too-high threshold can significantly decrease the efficiency of the algorithm. Once the number of non-dominated solutions exceeds the threshold, the distribution of particles in the  $D_i$  regions within the external archive is examined. In regions where multiple particles exist, only one particle is retained and the rest are deleted. This process is repeated until the maximum number of iterations is reached or non-dominated particles exist in all  $D_i$  regions.

### 3.3. Grouping Method Based on CRD Calculation and DD

#### 3.3.1. Calculation Method of CRD

In [26], a method for quantitatively calculating the CRD of decision variables is proposed. It involves the following steps:

(1) Normalize the objective space and generate the reference vector  $v_j$ ; (2) For each reference vector, its neighboring solutions are determined by  $Neighbor(T)$ , which represents the set of solutions closest to that reference vector among all candidate solutions. Therefore, it is necessary to calculate the angle between each candidate solution and the reference vector and select the  $T$  closest candidate solutions. The angle is calculated using the following formula:

$$angle(s, v_j) = arccos\left(\frac{F^T(s)v_j}{\|F(s)\|\|v_j\|}\right) \quad (20)$$

where  $s$  represents a candidate solution and  $F(s)$  represents the connecting vector between the candidate solution and the normalized origin; (3) For each of the  $T$  candidate solutions,  $k$  perturbations are added to each dimension and the generated sampled solutions are fitted into a line. The projection  $LN_{i,j,T}$  of these solutions on the reference vector is calculated. The computation of CRD is related to the length of the projection and the angle, and the specific calculation formula is as follows:

$$CRD_{i,j} = \left(1 + \frac{\theta_{i,j} - \theta_{min}}{\theta_{max} - \theta_{min}}\right) e^{-LN_{i,j}} \quad (21)$$

where  $\theta_{i,j}$  represents the angle between the fitted line and the reference vector and  $LN_{i,j}$  denotes the length of the projection of the fitted line on the reference vector.  $\theta_{min}$  and  $\theta_{max}$ , respectively, represent the minimum and maximum values among all the angles between candidate solutions and the reference vector, as shown in Figure 4. In the end, for each reference vector,  $T$  CRDs are calculated for each dimension and their average is computed as a measure of the convergence relevance for that dimension.

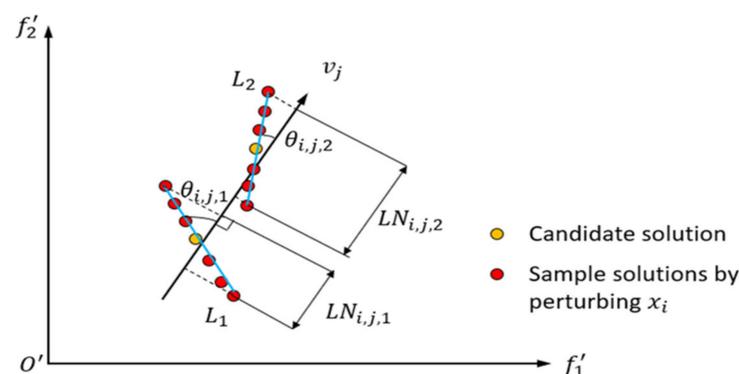


Figure 4. Schematic diagram of the parameters of the CRD calculation method ( $T = 2$ ).

### 3.3.2. Grouping Strategy Based on DD

After the calculation of CRD, the decision variables should be divided into multiple subgroups. The number of subgroups should not be too small relative to the number of decision variables to avoid having some subgroups with a large number of variables, which would result in ineffective grouping. Therefore, a strategy is proposed to group the variables based on the DD. This strategy automatically determines the number of subgroups and their sizes.

The CRD values of the decision variables are sorted in descending order, and the maximum value,  $crd_{max}$ , and the minimum value,  $crd_{min}$ , in this sequence are identified. Based on the predefined minimum number of subgroups,  $sub_{min}$ , the difference threshold is defined as follows:

$$DT = (crd_{max} - crd_{min}) / sub_{min}. \quad (22)$$

Then, the DD of adjacent values in the CRD sequence is calculated as follows:

$$DD_i = crd_i - crd_{i-1}, \quad (23)$$

where  $i$  represents the index of the CRD sequence. If  $DD_i \leq DT$ , the decision variable is placed in the current subgroup. If  $DD_i > DT$ , the decision variable is placed in a new subgroup. This process is repeated until all decision variables are grouped. According to the calculation principle of CRD, the smaller-numbered subgroups contribute more to diversity. The overall pseudocode for the adaptive grouping based on the CRD method is shown in Algorithm 1.

---

#### Algorithm 1: Framework based on CRD and DD grouping

---

**Input:**  $X$ : particles after initialization;  $fx1$  and  $fx2$ : objective function values  $RMSE$  and  $RBI$ ;  $N$ : number of particles;  $D$ : number of decision variables;  $xlimit$ : particle position boundary;  $sub_{min}$ : minimum number of groups.

1. Set the reference point and obtain the reference vector  $v_j$ .
2. Calculate the angle between each particle and each reference vector.
3. Select the two particles with the smallest angle as  $Neighbor(T)$ .
4. Add perturbation in each dimension, fit the generated sampling solution to a straight line, and calculate the angle and projection ( $\theta_{i,j}$  and  $LN_{i,j}$ ) between the straight line and the reference vector.
5. Calculate CRDs according to Formula (21).
6. Grouping decision variables based on DD.

**Output:** **Group\_num**: number of groups; **Subgroup**: decision variables within subgroups.

---

### 3.4. Algorithm Framework

The framework of IALSCC is described as follows (see Algorithm 2), and Figure 5 illustrates the flowchart of the proposed IALSCC algorithm for sensor layout optimization, which is divided into three steps: model building, variable grouping, and optimization.

**Algorithm 2:** Framework of IALSCC

**Input:**  $N_{ini}$ : number of initial particles;  $T_{ini}$ : number of initial particle iterations; **iter**: the maximum number of iterations;  $D$ : number of decision variables; **xlimit**: particle position boundary; **vlimit**: particle velocity boundary; **loop**: co-evolution times;  $\omega$ : inertia weight;  $c_1$  and  $c_2$ : self-awareness coefficient and group cognitive coefficient;  $D_j$ : number of divisions; **sub<sub>min</sub>**: minimum number of groups.

**Termination condition:** the maximum number of iterations is reached or region particle completeness is satisfied.

**Step 1: Initialization**

1. Using  $N_{ini}$  and  $T_{ini}$ , perform two single-objective optimizations on the two objective functions.
2. Grouping decision variables using CRD and DD.
3. Using the initial region division strategy, filter out the non-dominated solutions and save them in the external file and then obtain the number of regions (**space\_n**) that satisfy the particle existence condition.

**Step 2: Iteration**

For  $i \leftarrow 1$  to loop do

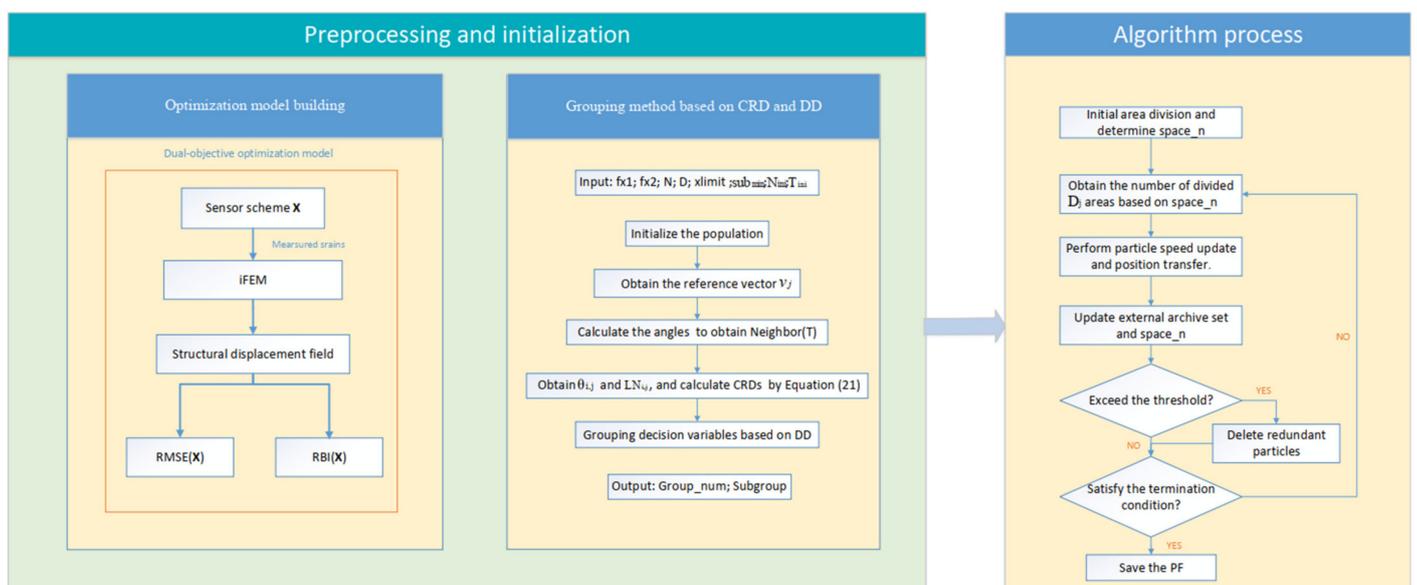
For  $j \leftarrow 1$  to Group\_num do

1. The number of divided areas  $D_j$  is determined by the range of **space\_n**.
2. Randomly select a particle in each region as the gbest of the region.
3. Perform particle speed update and position transfer.
4. The non-dominated solutions are screened out from the updated particles and added to the external archives, and the non-dominated solutions need to be screened again in the external archive.
5. Calculate the number of areas in the external file that satisfy the existence of particles and update **space\_n**.
6. Determine whether the particles in the external archive exceed the threshold and delete redundant particles if the threshold is exceeded.
7. Judging whether the termination condition is satisfied. If so, output the external archive; otherwise, enter the next iteration.

END

END

**Output:** external archive.



**Figure 5.** Algorithm Flowchart.

## 4. Algorithm Evaluation

### 4.1. Algorithm Effect Verification

To validate the optimization effectiveness of the algorithm, the grouping method proposed in this paper based on CRD and DD will be compared with other grouping methods, demonstrating the superiority of this grouping approach. The improved optimizer will be compared with AADMOPSO presented in [32], to demonstrate the effectiveness of the optimizer's improvements. The ZDT1, ZDT2, ZDT3, and ZDT6 test problems from the ZDT benchmark suite are selected as validation models. ZDT1 exhibits a continuous and uniformly distributed PF, ZDT2 features a non-convex PF, ZDT3 consists of five non-convex PFs, and ZDT6 has a non-uniformly distributed and non-convex PF [33]. The algorithm's convergence and diversity are evaluated using the minimum inverted generational distance (IGD) on the aforementioned test function instances with 1000 decision variables [34]. A smaller IGD value indicates better overall performance of the algorithm in terms of convergence and diversity.

As shown in Table 1, Algorithm 1 utilizes random grouping as the grouping method. Algorithm 2 employs linear grouping as the grouping method. Algorithm 3 adopts the grouping method based on CRD and DD and all three algorithms utilize the improved MOPSO as the optimizer. Algorithm 4 combines the grouping method based on CRD and DD with the AADMOPSO optimizer. For each algorithm, 20 independent runs were conducted on each test function and the computational results were statistically analyzed to obtain the mean and variance. Table 1 presents the average and standard deviation of the IGD values for the four algorithms solving the four test problems.

**Table 1.** The mean and standard deviation of the IGD values of the four algorithms on the four test functions.

Test Function	Grouping Method	Random Grouping	Linear Grouping	CRD and DD	CRD and DD
	Optimizer	Improved MOPSO	Improved MOPSO	Improved MOPSO	AADMOPSO
ZDT1	Mean	$2.668 \times 10^{-1}$	$8.803 \times 10^{-2}$	<b><math>8.628 \times 10^{-2}</math></b>	2.32
	Std	$2.376 \times 10^{-1}$	$8.394 \times 10^{-2}$	<b><math>4.967 \times 10^{-2}</math></b>	$6.377 \times 10^{-2}$
ZDT2	Mean	1.282	$6.916 \times 10^{-1}$	<b><math>4.603 \times 10^{-1}</math></b>	3.726
	Std	$8.706 \times 10^{-1}$	$4.881 \times 10^{-1}$	<b><math>1.828 \times 10^{-1}</math></b>	$1.839 \times 10^{-1}$
ZDT3	Mean	$1.887 \times 10^{-1}$	$6.280 \times 10^{-1}$	<b><math>1.085 \times 10^{-1}</math></b>	3.096
	Std	$7.045 \times 10^{-1}$	$1.106 \times 10^{-1}$	<b><math>2.311 \times 10^{-2}</math></b>	$1.189 \times 10^{-1}$
ZDT6	Mean	$8.818 \times 10^{-1}$	$8.793 \times 10^{-1}$	<b><math>8.789 \times 10^{-1}</math></b>	$8.798 \times 10^{-1}$
	Std	$1.723 \times 10^{-3}$	$1.487 \times 10^{-3}$	$1.579 \times 10^{-3}$	<b><math>5.541 \times 10^{-4}</math></b>

The data in the table reveal that the proposed IALSCC algorithm, in most test functions, exhibit smaller mean and standard deviation of IGD values compared to the other algorithms. This indicates that IALSCC demonstrates superior convergence and diversity, as well as better algorithm stability. Thus, the performance of the algorithm has been validated.

We verify the high efficiency of the proposed algorithm IALSCC from two aspects. On the one hand, we illustrate based on the data that choosing a suitable threshold value can improve the efficiency of the algorithm. For each threshold, the algorithm was tested 20 times independently under the test function and the average value of the data was selected as the test value. Table 2 gives the IGD and running time of IALSCC for different thresholds. On the other hand, we demonstrate that the grouping method of the CC framework improves the efficiency of the algorithm. We test the performance and efficiency of the IALSCC algorithm using the grouping method and only using the improved MOPSO. Each algorithm was run independently 20 times on different test functions and the average value was selected as the test value. Table 3 gives the IGD values and running time of both algorithms.

**Table 2.** The effect of different thresholds on algorithm performance and efficiency ( $d = 1000$ ).

IALSCC	Threshold	20	50	100	200
ZDT1	IGD	$2.191 \times 10^{-1}$	$1.274 \times 10^{-1}$	$8.628 \times 10^{-2}$	$6.617 \times 10^{-2}$
	Running time(s)	78.154	82.934	93.264	137.079
ZDT2	IGD	1.102	$7.317 \times 10^{-1}$	$4.603 \times 10^{-1}$	$3.217 \times 10^{-1}$
	Running time(s)	67.120	73.372	82.853	130.267
ZDT3	IGD	$2.387 \times 10^{-1}$	$2.199 \times 10^{-1}$	$1.085 \times 10^{-1}$	$9.681 \times 10^{-2}$
	Running time(s)	71.291	75.679	83.676	124.128
ZDT6	IGD	$8.834 \times 10^{-1}$	$8.812 \times 10^{-1}$	$8.789 \times 10^{-1}$	$8.770 \times 10^{-1}$
	Running time(s)	80.588	86.400	91.917	133.022

**Table 3.** Impact of grouping methods on algorithm performance and efficiency ( $d = 1000$ ).

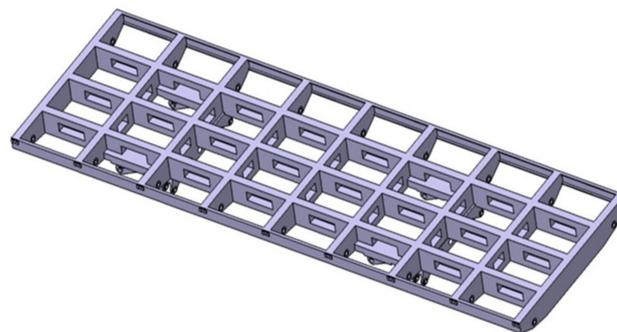
Optimization Algorithm	Test Function	ZDT1	ZDT2	ZDT3	ZDT6
IALSCC	IGD	$8.628 \times 10^{-2}$	$4.603 \times 10^{-1}$	$1.085 \times 10^{-1}$	$8.789 \times 10^{-1}$
	Running time(s)	93.264	82.853	83.676	91.917
Improved MOPSO	IGD	$8.648 \times 10^{-2}$	$3.059 \times 10^{-1}$	$9.167 \times 10^{-2}$	$7.977 \times 10^{-1}$
	Running time(s)	170.376	160.654	150.267	169.386

From Table 2, it can be seen that as the threshold increases, the running time required by the algorithm becomes longer; especially after the threshold is specified as 200, the running time increases significantly, but at the same time, the performance of the algorithm is not significantly improved, so we choose the threshold of 100 to improve efficiency and at the same time can ensure good performance.

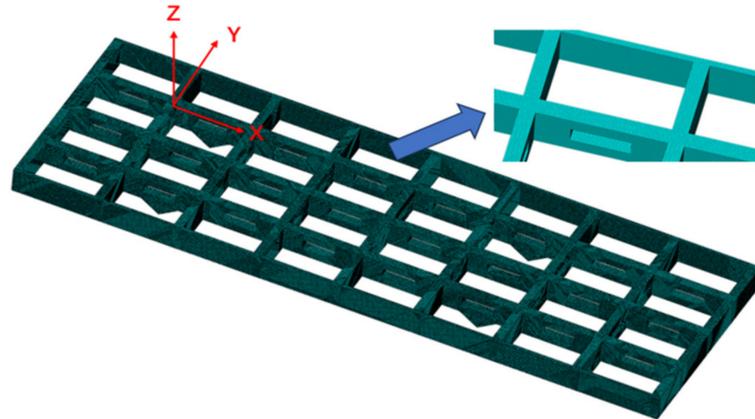
From Table 3, we can see that after using the grouping method in the CC framework, the performance of the two algorithms is not much different, but the efficiency of the IALSCC algorithms can be about doubled.

#### 4.2. Numerical Validation

To validate the generality of the IALSCC algorithm in the application of high-dimensional sensor placement, a simulation is conducted on a complex antenna truss structure. The finite element model of the antenna truss structure is established using the ABAQUS modeling and simulation software; it is modeled using 698,208 elements, including beam and solid. The Young's modulus is  $E = 7.03$  GPa, the Poisson's ratio is  $\nu = 0.3$ , and the density is  $\rho = 2700$  kg/m<sup>3</sup>. The overall dimensions of the antenna truss structure are 5944 mm  $\times$  1926 mm. The longitudinal beams consist of two side beams and three middle main beams. Both the main beams and side beams have a length of 653 mm. The main beams have a cross-sectional dimension of 260 mm  $\times$  80 mm, while the side beams have a cross-sectional dimension of 120 mm  $\times$  30 mm. The transverse beams have a length of 410 mm and a cross-sectional dimension of 260 mm  $\times$  80 mm. The hollow section in the main longitudinal beams has dimensions of 300 mm  $\times$  100 mm and is located at the center of the main beams, as shown in Figure 6.

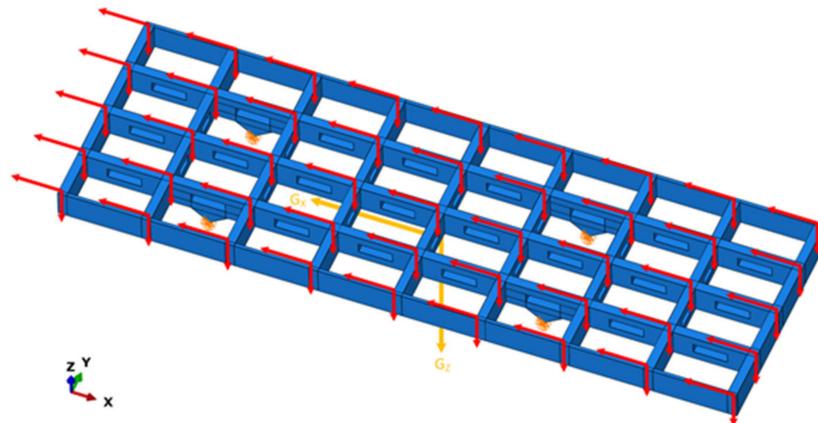
**Figure 6.** Antenna Truss Model.

The coordinate system and grid division of the entire antenna truss structure are shown in Figure 7, with a grid spacing of 10 mm. Each longitudinal beam of the truss structure is divided into eight segments by ribs, resulting in a total of 40 inverse finite-element beam elements. Each inverse finite element is equipped with six strain sensors, leading to a total of forty by six sensors required for the entire truss structure. Therefore, one set of sensor layout scheme consists of two hundred forty sensors.



**Figure 7.** Truss Meshing and Coordinate System.

One side of the antenna truss is fixed to the bottom through a hinge, while the other side is connected to an adjustable top rod through a hinge, which is used to adjust the angle of the antenna array. Therefore, the constraints of the truss can be simplified to four hinge constraints that constrain the displacement degrees of freedom of the structure and release the rotation degrees of freedom of the structure. The truss structure is subjected to the gravity of the structure itself and concentrated forces at each node. The force and constraint conditions of the truss structure are shown in Figure 8. When the truss structure is tilted and the angle is changed, the distribution of gravity and concentrated forces in each direction will also change accordingly. Three typical operating conditions are selected for force analysis and the force conditions are shown in Table 4.

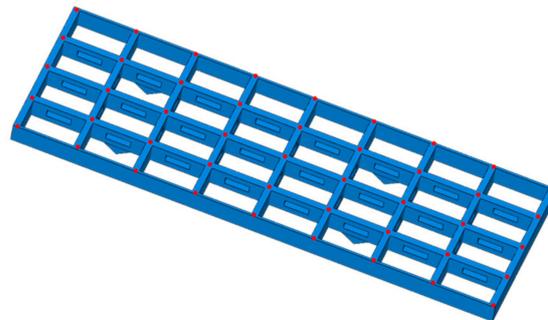


**Figure 8.** Forces and constraints of truss structures.

**Table 4.** The load applied by the antenna array (N).

Load	10°	45°	80°
Gravity component in X direction	−1700.898	−6926.89	−9649.91
Gravity component in Z direction	−9651.267	−6932.41	−1708.58
Concentration force in X direction	−4512.6	−18,377.5	−25,601.8
Concentration force in Z direction	−25,605.4	−18,392.1	−4532.98

The strain information of nodes is extracted for three working conditions and the mapping relationship between sensor locations and the accuracy and robustness indicators is established based on the iFEM mentioned in Chapter 2. A high-dimensional bi-objective sensor layout optimization model is constructed. The validation points are uniformly selected throughout the truss structure, as shown in Figure 9. Additionally, since the primary deformation directions are X and Z, the accuracy indicator is calculated as the average RMSE in both directions.

**Figure 9.** Check point distribution.

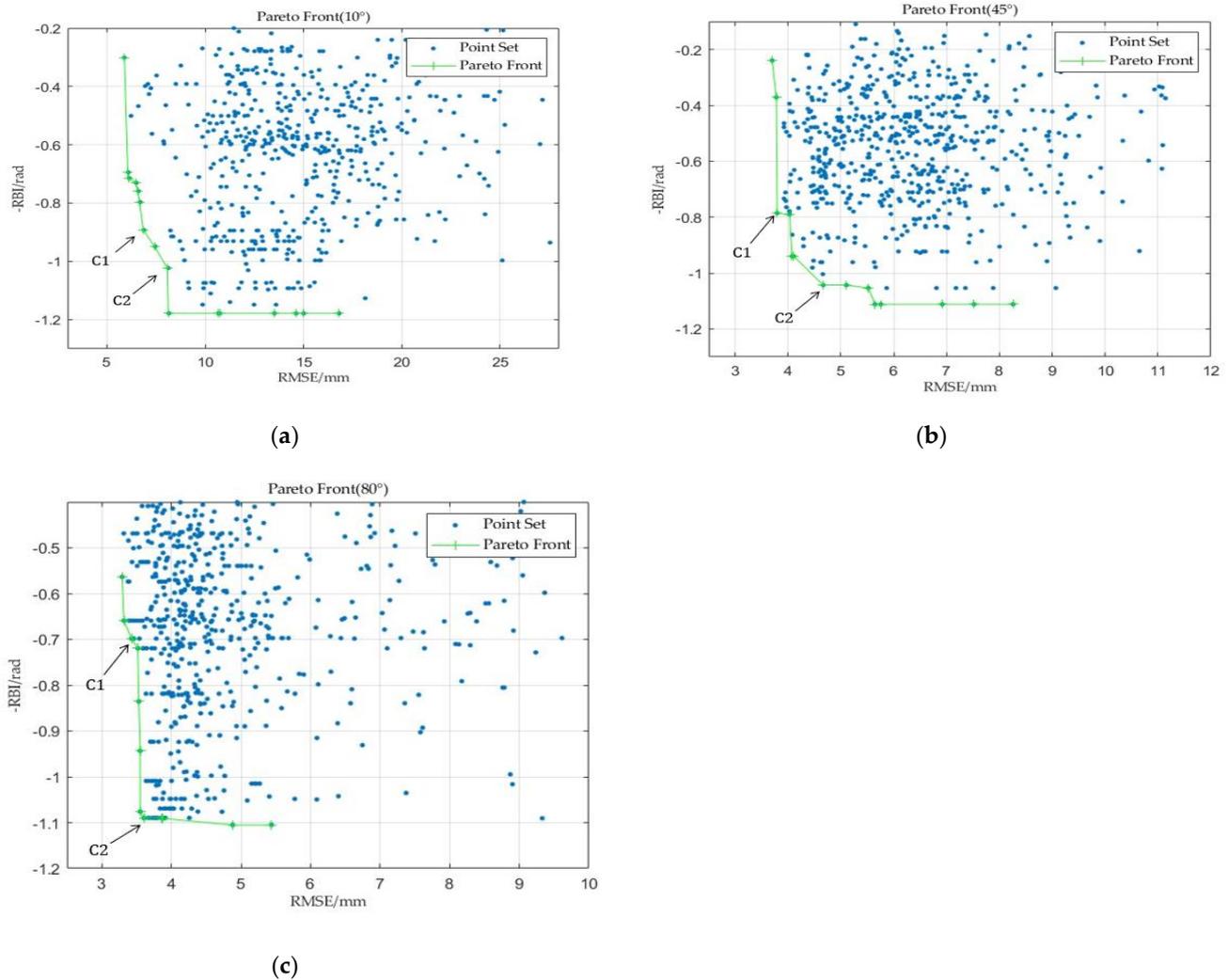
The proposed IALSCC algorithm is applied for sensor layout optimization in the high-dimensional truss model, where the objective space consists of a series of nodes carrying strain information. The parameters are set as follows: the minimum number of groups  $sub_{min} = 10$ , the maximum number of cycles of co-evolution is set to  $loop = 10$ , the maximum number of subpopulation iterations is set to  $iter = 60$ , and the number of divisions  $Di = 40$ .

The optimization results are shown in Figure 10, where the blue particles represent the positions explored by the population and the green front represents the PF composed of non-dominated solutions. To visually display the PF, the figure only shows the solutions in the vicinity of the explored PF region. Each point in the figure contains the sensor position information under this scheme. From the PF shown in the figure, it can be inferred that the optimized sensor layout solution can achieve a high reconstruction accuracy.

Subsequently, typical solutions C1 and C2 are selected from the PF of the three operating conditions for the analysis of reconstruction accuracy and robustness. To visually demonstrate the reconstruction performance, the maximum displacement error  $ERR^{max}$  is introduced, which is defined as follows:

$$ERR^{max} = \max \left| \text{disp}_i^{\text{measurement}}(\mathbf{X}) - \text{disp}_i^{\text{iFEM}}(\mathbf{X}) \right|, i = 1, \dots, 45, \quad (24)$$

where  $\text{disp}_i^{\text{measurement}}$  represents the reference displacement of the  $i$ -th calibration point, extracted from the simulation software, and  $\text{disp}_i^{\text{iFEM}}$  represents the displacement of the  $i$ -th calibration point reconstructed using the iFEM. Table 5 shows the maximum displacement error corresponding to the two sensor layout schemes and  $\text{Max\_disp}$  is the maximum deformation.



**Figure 10.** (a) PF of  $10^\circ$  working condition; (b) PF of  $45^\circ$  working condition; (c) PF of  $80^\circ$  working condition.

**Table 5.** The maximum displacement error of a typical scheme (mm).

Condition	Direction	Max_disp	C1	C2
$10^\circ$	X	4.9752	1.9373	2.2805
	Z	−55.6114	7.0224	8.3916
$45^\circ$	X	−7.2117	1.2756	1.8588
	Z	−30.2659	4.3316	4.8450
$80^\circ$	X	−10.0236	3.0336	3.6807
	Z	−11.4974	3.1346	3.6880

Comparing the maximum displacement errors between the two schemes, under the condition of no interference when the antenna framework is tilted by  $10^\circ$  with the Z direction being the primary deformation direction and a maximum deformation of  $-55.6114$  mm, the maximum displacement error for sensor layout scheme C1 is 7.0224 mm, which is better than the maximum displacement error of 8.3916 mm for sensor layout scheme C2. The maximum deformation in the X direction is 4.9752 mm, which can be considered negligible compared to the Z direction. When the antenna framework is tilted by  $45^\circ$  with maximum deformations of  $-7.2117$  mm and  $-30.2659$  mm in the X and Z directions, respectively, the maximum displacement errors for both directions under scheme C1 are 1.2756 mm and 4.3316 mm, which are superior to the maximum displacement errors of 1.8588 mm and 4.8450 mm under scheme C2. When the antenna framework is tilted by

80° with maximum deformations of  $-10.0236$  mm and  $-11.4974$  mm in the X and Z directions, respectively, the maximum displacement errors for both directions under scheme C1 are 3.0336 mm and 3.1346 mm, also outperforming the maximum displacement errors of 3.6807 mm and 3.6880 mm under scheme C2. The results demonstrate that the sensor layout scheme C1 exhibits higher reconstruction accuracy compared to scheme C2, consistent with the distribution of the reconstruction accuracy indicator *RMSE*, further reflecting the effectiveness of the reconstruction metrics.

In order to observe the stability of the two sensor layout schemes, 500 random disturbances are added to each sensor independently within the error range as specified in Equation (25); the class stiffness matrix will be recalculated after each addition of disturbances as it is only related to the sensor position, and then we perform iFEM reconstruction to evaluate the reconstruction error at this point. Finally, the data with the largest reconstruction error among the 500 experiments is selected and compared with Table 3 to verify the robustness metrics. The average maximum displacement error from 500 simulations is recorded and is presented in Table 4.

$$\Delta x_i \in (-0.3, 0.3), \Delta y_i \in (-0.05, 0.05), \Delta z_i \in (-0.05, 0.05), i = 1, \dots, 240; \quad (25)$$

According to Table 6, when the framework is lifted at 10° and disturbances are added, the maximum displacement errors in both directions for the C1 scheme increased from 1.9373 mm and 7.0224 mm to 2.7534 mm and 9.7508 mm, respectively, with a change of 0.8161 mm and 2.7284 mm. On the other hand, for C2 scheme the maximum displacement errors increase from 2.2805 mm and 8.3917 mm to 2.6112 mm and 9.0987 mm, respectively, with a change of 0.3307 mm and 0.707 mm. The data indicate that after introducing disturbances, the resistance ability of C2 scheme against disturbances is superior to that of C1 scheme and it also exhibits better reconstruction accuracy. Similar conclusions can be drawn for other working conditions.

**Table 6.** The average maximum displacement error after adding 500 disturbances (mm).

Condition	Scheme	X	Z
10°	C1	2.7534	9.7508
	C2	2.6112	9.0987
45°	C1	2.3212	5.6491
	C2	2.4418	5.4987
80°	C1	4.1647	4.2683
	C2	4.3429	4.2425

When selecting a sensor scheme, it is necessary to consider the model's requirements for both reconstruction accuracy and stability. Given that the antenna needs to operate under complex working conditions, it is more appropriate to choose sensor layout scheme C2, which exhibits better resistance against disturbances, while still meeting the accuracy requirements.

In conclusion, the algorithm proposed in this paper can be used to solve the problem of optimal sensor layout for complex structures, thus enabling health monitoring of large structures, such as antenna trusses, large aircraft wings, large bridges, and other engineering structures. When carrying out health monitoring of large structures, there is a greater demand for hardware demodulation capability and sensor failure detection capability, which is also a greater difficulty, and the pressure on the performance of the algorithm is also enormous if the inverse finite element is not reasonably divided in the face of a very complex structure. Our next research steps are as follows: (1) A more extensive study of grouping strategies in the CC framework; (2) Regarding the collaboration method within the CC framework, whether there is any space that can be further improved; (3) Whether there is a more reasonable method for threshold determination. An ongoing problem in the development of high-dimensional optimization algorithms is that it is difficult to

simultaneously improve algorithmic performance and efficiency, and future research will also be devoted to solving this problem.

## 5. Conclusions

An optimization algorithm for sensor layout optimization problems in complex structures is proposed in this paper. Firstly, an inverse finite element model for multi-objective optimization of sensor layout is established based on reconstruction accuracy and robustness indicators. To address the issues of falling into local optima and low algorithm efficiency in previous algorithms when facing high-dimensional sensor optimization layout models, an IALSCC algorithm is proposed. It incorporates initialization strategies, adaptive region partitioning strategies, *gbest* selection, and particle update strategies to enhance algorithm diversity and introduces grouping methods based on the calculation of CRD and DD to improve algorithm efficiency. Then, the performance and efficiency of the IALSCC algorithm are verified in different comparative experiments. Finally, the proposed optimization algorithm IALSCC is applied to the deformation reconstruction of a complex antenna truss model, and the simulation results demonstrate its effectiveness in handling high-dimensional sensor layout optimization problems.

**Author Contributions:** Conceptualization, Z.Z. and H.B.; methodology, Z.Z.; software, Y.L.; validation, Z.Z., K.C. and Y.L.; formal analysis, Z.Z.; investigation, Z.Z.; resources, Z.Z.; data curation, Z.Z.; writing—original draft preparation, Z.Z.; writing—review and editing, Z.Z.; visualization, Z.Z.; supervision, Z.Z.; 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 the National Natural Science Foundation of China, grant numbers: 51775401 and 52275542.

**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. Hassan, S.; Ka-Veng, Y. Structural health monitoring by a novel probabilistic machine learning method based on extreme value theory and mixture quantile modeling. *Mech. Syst. Signal Process.* **2022**, *173*, 109049.
2. Xiao, F.; Sun, H.; Mao, Y.; Chen, G.S. Damage identification of large-scale space truss structures based on stiffness separation method. *Structures* **2023**, *53*, 109–118. [[CrossRef](#)]
3. Xiao, F.; Zhu, W.W.; Meng, X.W.; Chen, G.S. Parameter Identification of Frame Structures by considering Shear Deformation. *Int. J. Distrib. Sens. Netw.* **2023**, 6631716. [[CrossRef](#)]
4. Xiao, F.; Zhu, W.W.; Meng, X.W.; Chen, G.S. Parameter identification of structures with different connections using static responses. *Appl. Sci.* **2022**, *12*, 5896. [[CrossRef](#)]
5. Gherlone, M.; Cerracchio, P.; Mattone, M.; Disciupa, M.; Tessler, A. Shape sensing of 3D frame structures using an inverse Finite Element Method. *Int. J. Solids Struct.* **2012**, *49*, 3100–3112. [[CrossRef](#)]
6. Zhao, Y.; Bao, H.; Duan, X.C.; Fang, H.M. Research of Situ Strain of Measuring Flexible Truss Deformation Based on Fuzzy Network Method. In Proceedings of the 2016 International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2016), Guilin, China, 24–25 September 2016.
7. Ma, T.S.; Kuang, P.; Tian, W.H. An improved recurrent neural network for 3D object reconstruction. *Appl. Intell.* **2020**, *50*, 905–923. [[CrossRef](#)]
8. Kim, H.I.; Kang, L.H.; Han, J.H. Shape estimation with distributed fiber Bragg grating sensors for rotating structures. *Smart Mater. Struct.* **2011**, *20*, 035011. [[CrossRef](#)]
9. Rapp, S.; Kang, L.H.; Han, J.H. Displacement field estimation for a two-dimensional structure using fiber Bragg grating sensors. *Smart Mater. Struct.* **2009**, *18*, 025006. [[CrossRef](#)]
10. Jineesh, T.; Gurusamy, S.; Rajanna, T.R.; Sundarrajan, A. Structural Shape Estimation by Mode Shapes Using Fiber Bragg Grating Sensors: A Genetic Algorithm Approach. *IEEE Sens. J.* **2020**, *20*, 2945–2952.
11. Zhu, M.; Michael, T. *Comparison of Shape Reconstruction Strategies in a Complex Flexible Structure*; University of California: San Diego, CA, USA, 2008; p. 6932.

12. Tessler, A.; Spangler, J.L. *Inverse FEM for Full-Field Reconstruction of Elastic Deformations in Shear Deformable Plates and Shells*; DEStech Publications, Inc.: Lancaster, PA, USA, 2004.
13. 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](#)]
14. Gherlone, M.; Cerracchio, P.; Mattone, M. An inverse finite element method for beam shape sensing: Theoretical framework and experimental validation. *Smart Mater. Struct.* **2014**, *23*, 045027. [[CrossRef](#)]
15. Cerracchio, P.; Gherlone, M.; Tessler, A. Real-time displacement monitoring of a composite stiffened panel subjected to mechanical and thermal loads. *Meccanica* **2015**, *50*, 2487–2496. [[CrossRef](#)]
16. 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](#)]
17. Chen, K.Y.; He, D.H.; Zhao, Y. A unified full-field deformation measurement method for beam-like structure. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–10. [[CrossRef](#)]
18. Xiao, F.; Hulsey, J.L.; Chen, G.S.; Xiang, Y. Optimal static strain sensor placement for truss bridges. *Int. J. Distrib. Sens. Netw.* **2017**, *13*, 155014771770792. [[CrossRef](#)]
19. Zhao, Y.; Du, J.L.; Bao, H.; Xu, Q. Optimal Sensor Placement for Inverse Finite Element Reconstruction of Three-Dimensional Frame Deformation. *Int. J. Aerosp. Eng.* **2018**, *2018*, 6121293. [[CrossRef](#)]
20. Zhao, F.F.; Bao, H.; Xue, S.; Xu, Q. 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](#)]
21. Li, X.H.; Niu, S.T.; Bao, H.; Hu, N.G. Improved Adaptive Multi-Objective Particle Swarm Optimization of Sensor Layout for Shape Sensing with Inverse Finite Element Method. *Sensors* **2022**, *22*, 5203. [[CrossRef](#)]
22. Potter, M.A.; De Jong, K.A. A cooperative coevolutionary approach to function optimization. In *International Conference on Parallel Problem Solving from Nature*; Springer: Berlin/Heidelberg, Germany, 1994; pp. 249–257.
23. Omidvar, M.N.; Li, X.; Yang, Z. Cooperative co-evolution for large scale optimization through more frequent random grouping. In *IEEE Congress on Evolutionary Computation*; IEEE: New York, NY, USA, 2010; pp. 1–8.
24. Van, A.S.; Wang, X.S.; Zamar, R.H. Linear grouping using orthogonal regression. *Comput. Stat. Data Anal.* **2006**, *50*, 1287–1312.
25. Omidvar, M.N.; Li, X.; Yao, X. Cooperative co-evolution with delta grouping for large scale non-separable function optimization. In *IEEE Congress on Evolutionary Computation*; IEEE: New York, NY, USA, 2010; pp. 1–8.
26. Zhang, X.Y.; Gong, Y.J.; Lin, Y. Dynamic cooperative coevolution for large scale optimization. *IEEE Trans. Evol. Comput.* **2019**, *23*, 935–948. [[CrossRef](#)]
27. Li, M.; Wei, J. A cooperative co-evolutionary algorithm for large-scale multi-objective optimization problems. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, Kyoto, Japan, 15–19 July 2018; pp. 1716–1721.
28. Ma, L.; Huang, M.; Yang, S. An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization. *IEEE Trans. Cybern.* **2021**, *52*, 6684–6696. [[CrossRef](#)] [[PubMed](#)]
29. Gordis, J.H.; Shin, J.C.; 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](#)]
30. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Proceedings of the ICNN'95-International Conference on Neural Networks*, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
31. Coello, C.A.C.; Lechuga, M.S. MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation*. CEC'02 (Cat. No. 02TH8600), Honolulu, HI, USA, 12–17 May 2002; pp. 1051–1056.
32. Feng, Q.; Li, Q.; Chen, P. Multiobjective particle swarm optimization algorithm based on adaptive angle division. *IEEE Access* **2019**, *7*, 87916–87930. [[CrossRef](#)]
33. Zitzler, E.; Deb, K.; Thiele, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.* **2000**, *8*, 173–195. [[CrossRef](#)] [[PubMed](#)]
34. Sierra, M.R.; Coello, C.C. A new multi-objective particle swarm optimizer with improved selection and diversity mechanisms. Technical Report of CINVESTAV-IPN. 23 November 2004.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.