



ARTICLE

Interior Point Method-Assisted Differential Evolution for Expensive Optimization of Secondary Source Deployment

Chao Wu¹ and Weijian Kong^{2,*}

¹School of Information Science and Technology, Donghua University, Shanghai, China

²Engineering Research Center of Digitized Textile & Apparel Technology, Ministry of Education, Donghua University, Shanghai, China

*Corresponding Author: Weijian Kong. Email: kongweijian@dhu.edu.cn

Received: 13 December 2025; Accepted: 26 January 2026; Published: 26 February 2026

ABSTRACT: In active noise control, the optimal deployment of secondary sources is a critical factor influencing the noise reduction performance due to the spatial inhomogeneity of the sound field. Traditional methods, which rely on finite element analysis to model the sound field, are accurate but computationally intensive, leading to high costs in solving the deployment optimization problem. To address this issue, this paper proposes an expensive optimization method for secondary source deployment based on Interior Point Method-assisted Differential Evolution with Weibull distribution (IPMDEW). During the optimization process, a Kriging model is employed to construct a response surface, i.e., a surrogate model, of the objective function. The surrogate model is used for the initial evaluation of the population, while the finite element model is utilized to verify promising individuals. A surrogate model update algorithm based on k-means clustering is designed to iteratively refine the model and enhance its accuracy. The IPMDEW algorithm utilizes the Weibull distribution-based weighted differential evolution for global exploration and switches to the gradient-based interior point method for refined local optimization when the population approaches convergence. The results demonstrate Kriging surrogate-assisted optimization method for secondary source deployment reduces the optimization time by 85.79%, i.e., by 347.64 h, significantly improving optimization efficiency. Furthermore, the accuracy of the Kriging model continuously improves during the optimization process. The proposed method achieves a noise reduction of 58.32 dB, ensuring high optimization accuracy while substantially increasing efficiency.

KEYWORDS: Spatial noise reduction; expensive optimization; active noise control (ANC); multi-channel

1 Introduction

In the study of active noise control (ANC), particularly in multi-channel systems, the deployment of secondary sources significantly influences the noise reduction performance. Researchers have employed non-deterministic methods, such as enumeration and sensitivity analysis [1,2], to determine the positions of secondary sources, which have, to some extent, addressed the deployment issue. However, in complex, open-field environments, these methods demonstrate limited optimization effectiveness and precision. Modeling the sound field using Finite Element Analysis (FEA) provides a foundation for optimizing secondary source deployment by enabling accurate calculation of the sound pressure levels in the target region based on a given configuration. While the finite element model is accurate, it is computationally intensive. Achieving effective spatial noise reduction necessitates deploying a sufficient number of secondary sources to cover the target area. As the number of secondary sources increases, the dimensionality of the optimization problem's

decision variables grows rapidly, and the computational complexity of calculating the sound pressure in the target region rises exponentially.

Surrogate-assisted nature-inspired algorithms demonstrate significant advantages in solving expensive optimization problems [3]. The general approach involves constructing a low-cost surrogate model and leveraging its synergy with the global search capability of a nature-inspired algorithm. The computational cost associated with these evaluations can be substantially reduced by collaboratively using the surrogate model and the complex physical (mechanistic) model. To overcome limitations of poor model accuracy and low computational efficiency associated with a single training data strategy in high-dimensional scenarios, various advanced methods have been proposed. Guo et al. introduced a classifier-based distributed surrogate-assisted evolutionary algorithm [4] for distributed data-driven optimization problems. Zhou et al. [5] employed a two-level surrogate model structure and achieved adaptive resource allocation through a dynamic management mechanism. Liu et al. [6] proposed a fast generalized surrogate-assisted evolution algorithm based on a multi-strategy hybrid sparrow search algorithm (FGSAEA-MSHSSA), which, by introducing an innovative model management framework, significantly reduces the number of function evaluations and enhances optimization performance. Rahi et al. [7] proposed a partial-evaluation-based strategy for expensive constrained optimization problems, which sequentially evaluates constraints according to their likelihood of violation and aborts evaluation upon detecting a violation. This approach, combined with an improved ranking mechanism, significantly reduces evaluation cost. Zhang et al. [8] proposed a hierarchical surrogate-assisted optimization algorithm, which locates promising regions during the global search phase and employs a dynamically ensemble surrogate model in the local search phase to accelerate convergence. Pan et al. [9] proposed a hybrid surrogate-assisted optimization algorithm that integrates global exploration and local exploitation, incorporating novel pre-screening and dual control strategies, and employs local RBF models to efficiently solve expensive optimization problems.

Among various heuristic algorithms, Differential Evolution (DE) offers potential for extension to expensive optimization problems due to its lower space complexity compared to many other Evolutionary Algorithms (EA) [10]. Initially proposed by Storn and Price in 1997 [11], the DE algorithm has since been significantly improved by researchers. Notable enhancements include: Farda et al. [12], who integrated adaptive mutation operators with a stochastic learning-based population improvement strategy; and Liu et al. [13], who proposed a learnable evolutionary framework based on reinforcement learning, enabling autonomous adaptation of algorithm parameters and intelligent optimization of search strategies through deep reinforcement learning.

In distributed active noise control systems, increasing the number of secondary sources can enhance global noise attenuation performance. However, this also increases the parameter dimensionality due to the inclusion of spatial coordinates and initial phase parameters for each source. This paper formulates the co-optimization of secondary source placement and initial phases as a high-dimensional constrained non-convex nonlinear programming problem. To address the non-convexity of the model, our previous work proposed the IPMDEW algorithm, which integrates the global exploration capability of an improved differential evolution with the local optimization of an interior point method, achieving effective solution of the problem under a non-surrogate framework. However, this algorithm requires multiple calls to computationally expensive finite element simulations during iteration. To improve optimization efficiency, this paper further introduces a surrogate model, constructing a Kriging surrogate model for calculating the superimposed sound pressure of secondary sources, and designs an iterative update mechanism for the surrogate model based on the k-means clustering method to ensure its reliability. The proposed expensive optimization algorithm employs a dynamic switching strategy between global stochastic search and local gradient-based optimization, significantly improving search efficiency in high-dimensional decision

spaces while balancing convergence speed and accuracy. Furthermore, by applying k-means clustering to dynamically select representative candidate solutions for finite element evaluation, the method effectively avoids frequent, computationally intensive FEA.

The structure of this paper is organized as follows: [Section 1](#) describes the optimization problem of secondary source deployment. [Section 2](#) presents the proposed Kriging surrogate-assisted Interior Point Method-assisted Differential Evolution with Weibull distribution (IPMDEW) method. [Section 3](#) presents the simulation experiments to validate the accuracy improvement of the Kriging model during the iterative process and the optimization effectiveness of the IPMDEW algorithm. Finally, [Section 4](#) concludes with a summary and outlines directions for future research.

2 Proposed Approach

The proposed approach utilizes the Kriging surrogate model and incorporates hybrid differential evolution with clustering-based iterative model updates to address efficiency challenges in expensive optimization problems.

2.1 Description of the Secondary Source Optimization Problem

Active Noise Control is based on the principle of destructive interference of sound waves. In an ideal fluid medium, neglecting the reflection of spherical waves, the sound pressure at any point in space can be expressed as:

$$p(r) = \frac{A}{r} e^{j(\omega t - kr + \varphi)} \quad (1)$$

where, r denotes the distance from the target point to sound source; $k = \frac{\omega}{c_c}$, which is known as the wave number; ω represents the angular frequency of the sound wave; $\varphi \in [0, \pi)$, which represents the initial phase of the sound source.

By defining the field point as the coordinate origin and taking into account the attenuation effect of each sound source, the resultant sound pressure at the field point derived from the principle of coherent wave superposition can be expressed as:

$$p_{total}(r) = \sum_{i=0}^n \frac{A_i}{r_i} e^{j(\omega t - kr_i + \varphi_i)} = A_{total} \cos(\omega t + \varphi_{total}) \quad (2)$$

Typically, the sound pressure level (SPL) is used to evaluate the noise reduction performance, where a lower SPL indicates better performance. The SPL function can be defined as:

$$f_{SPL}(X, \theta) = 20 \log_{10} \left(\frac{P_{rms}}{P_{ref}} \right) \quad (3)$$

$$P_{rms} = \sqrt{\frac{1}{T} \int_0^T \text{Re} \left\{ \sum_{i=0}^n \frac{A_i}{r_i} e^{j(\omega t - kr_i + \varphi_i)} \right\}^2 dt} \quad (4)$$

where, $X = (x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_m, y_m, z_m)$, $\theta = (\varphi_1, \varphi_2, \dots, \varphi_m)$, P_{ref} represents the reference sound pressure, set to 10^{-5} Pa, and P_{rms} denotes the effective sound pressure. $T = 1/f$, represents the frequency of the sound wave, set to 100 Hz.

Finally, the optimization problem can be formulated as follows:

$$\begin{cases} \min_X f(X) \\ h(x) = 0 \\ L \leq X \leq U \end{cases} \quad (5)$$

where $h(x)$ represents the boundary condition, which are handled using a Perfectly Matched Layer (PML), with related proofs available in Ref. [14]; L and U denote the lower and upper bounds of the decision variables, respectively.

2.2 Construction of the Kriging Model

For a given dataset $X = \{x^1, x^2, \dots, x^n\}^T$ with corresponding objective function values $Y = \{y^1, y^2, \dots, y^n\}$, it is assumed that all data follow an n -dimensional normal distribution, i.e., $(Y(x^1) Y(x^2) \dots Y(x^n))^T(\mu, \sigma^2)$. By taking the mean as a constant $\mathbf{1}\mu$, Where $\mathbf{1}$ is an $n \times 1$ matrix, the covariance C in the multidimensional random process is defined as [15]:

$$C = \begin{pmatrix} \text{cor}(Y(x^1), Y(x^1)), & \dots, & \text{cor}(Y(x^1), Y(x^n)) \\ \vdots & \ddots & \vdots \\ \text{cor}(Y(x^n), Y(x^1)), & \dots, & \text{cor}(Y(x^n), Y(x^n)) \end{pmatrix} \quad (6)$$

Based on the mean and variance, the conditional probability for a given value y can be expressed as:

$$L(y|\mu, \sigma) = \frac{1}{(2\pi\sigma^2)^{n/2}|C|^{1/2}} \exp\left(-\frac{(y - \mathbf{1}\mu)^T C^{-1}(y - \mathbf{1}\mu)}{2\sigma^2}\right) \quad (7)$$

To ensure the obtained conditional probability is maximized, the maximum likelihood estimation method is employed to compute the partial derivatives with respect to μ and σ sequentially. The resulting prior parameters $\hat{\mu}$ and $\hat{\sigma}^2$ are given by:

$$\hat{\mu} = \frac{\mathbf{1}^T C^{-1} y}{\mathbf{1}^T C^{-1} \mathbf{1}} \quad (8)$$

$$\hat{\sigma}^2 = \frac{(y - \mathbf{1}\hat{\mu})^T C^{-1}(y - \mathbf{1}\hat{\mu})}{n} \quad (9)$$

Next, prediction is performed using the aforementioned Kriging model. Assuming a prediction point \hat{y} , a new dataset $\tilde{y} = \{y^1, y^2, \dots, y^n, \hat{y}\}$ is formed by incorporating this point into the original sample data. The maximum likelihood function can be approximated as:

$$\ln L \approx -\frac{\begin{pmatrix} y - \mathbf{1}\hat{\mu} \\ \hat{y} - \hat{\mu} \end{pmatrix}^T \begin{pmatrix} C & r \\ r^T & 1 \end{pmatrix} \begin{pmatrix} y - \mathbf{1}\hat{\mu} \\ \hat{y} - \hat{\mu} \end{pmatrix}}{2\hat{\sigma}^2} \quad (10)$$

where, r represents the covariance matrix between data point x and prediction point \hat{x} , and by defining

$$\tilde{C} = \begin{pmatrix} C & r \\ r^T & 1 \end{pmatrix}.$$

By taking the partial derivative of Eq. (10), the predicted value \hat{y} is given by:

$$\hat{y}_1(x) = \hat{\mu} + r^T C^{-1}(y - \mathbf{1}\hat{\mu}) \quad (11)$$

2.3 The Kriging Surrogate-Assisted Optimization Method

Given the challenges of computational complexity and cost in the secondary source deployment optimization problem, this study proposes an optimization framework based on Kriging surrogate assistance, which integrates the IPMDEW algorithm. The proposed strategy achieves an effective balance between global exploration and local search capabilities, as illustrated in Fig. 1.

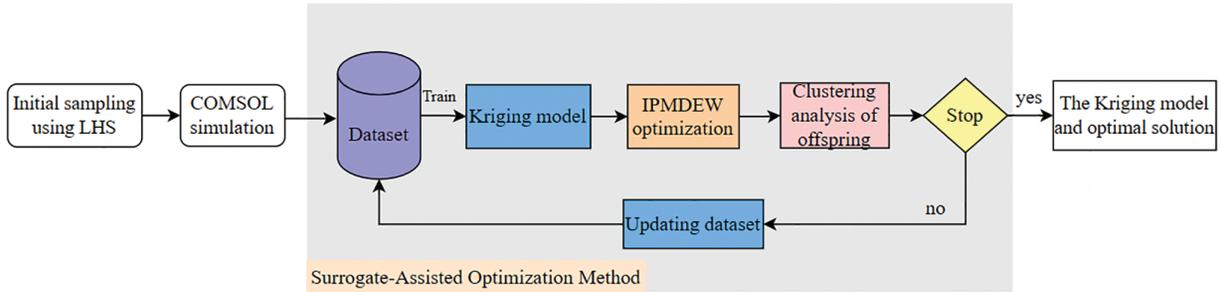


Figure 1: Framework of the Kriging surrogate optimization algorithm.

During the initialization phase, a high-information-density initial sample set is constructed via design of experiments, combining Latin Hypercube Sampling [16] (LHS) with historical datasets. Throughout the iterative optimization process, the Kriging model approximates the objective response values in real time to reduce computational overhead. To overcome the limitations of traditional passive surrogate modeling, an active learning mechanism is proposed: in each iteration, k-means clustering is applied to the offspring population generated by differential evolution. This method screens individuals based on diversity, selecting the most spatially representative candidate solutions for Finite Element validation. This strategy ensures that newly added sample points effectively probe undersampled design subspaces and mitigates the local overfitting of the surrogate model that can result from sample clustering.

2.4 The IPMDEW Algorithm

To optimize high-dimensional Kriging models, this paper proposes an integrated algorithm that combines the IPMDEW method (Algorithm 1). The algorithm employs the Weibull distribution-based Weighted Differential Evolution (MDEW) for global exploration. When the population converges, it switches to the Interior Point Method (IPM) for local refinement within the neighborhood of the best solution found. The detailed procedure is illustrated in Fig. 2.

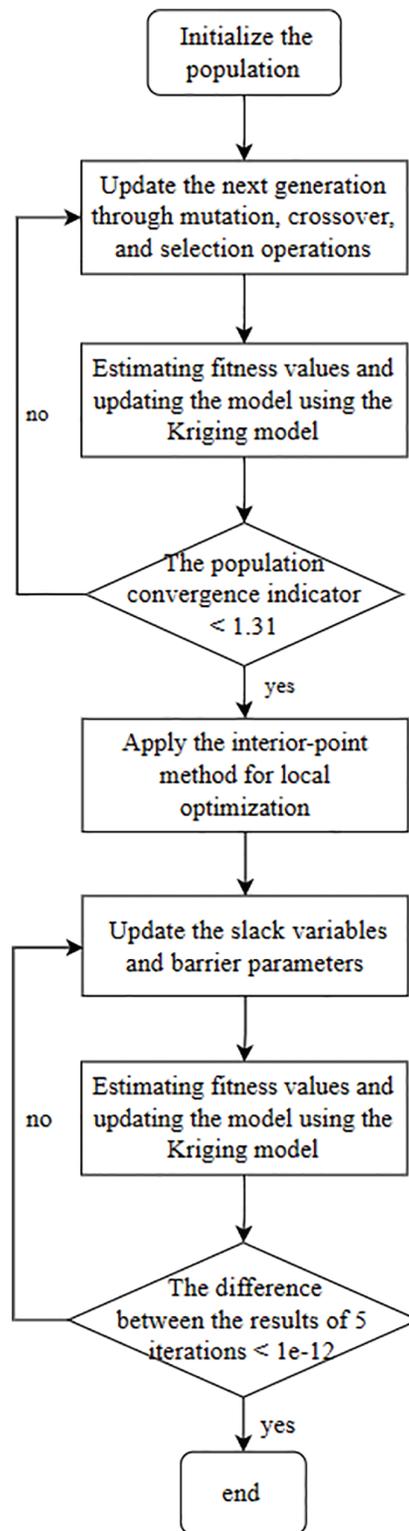


Figure 2: Flowchart of the IPMDEW algorithm.

Algorithm 1: IPMDEW

```

1: procedure MAIN
2:   model  $\leftarrow$  load MPH model
3:   Initialize parameters: dim, CR, pop_size, MAX_FES, MAX_ITE
4:   lb, ub  $\leftarrow$  problem bounds
5:   range  $\leftarrow$  ub - lb
6:   Generate initial population via Latin Hypercube Sampling
7:   Evaluate initial population using true objective function
8:   Initialize Best_agent and Best_fit
9:   Build initial Kriging model with initial samples
10:  FES  $\leftarrow$  1
11:  while FES  $\leq$  MAX_FES do
12:    for each individual  $j$  in population do
13:      if (rand  $\leq$  1 and FES  $\leq$   $0.3 \times$  MAX_FES) or (rand  $\leq$  0.4 and FES  $\leq$   $0.7 \times$  MAX_FES)
14:        then
15:          Apply WeibullFlight to generate  $X_{new}$ 
16:          Apply boundary constraints to  $X_{new}$ 
17:          Predict fitness using Kriging model
18:          Update individual if improved
19:        end if
20:      if rand  $\leq$  1 and FES  $\geq$   $0.3 \times$  MAX_FES then
21:        Change mutation strategy
22:      if FES  $\leq$   $0.8 \times$  MAX_FES then
23:        Apply mean mutation strategy
24:      else
25:        Apply deep intensification strategy
26:      end if
27:      Apply crossover operation
28:      Predict fitness using Kriging model
29:    end if
30:  end for
31:  Selectively calculate the true fitness values and update the Kriging model
32:  Calculate  $\delta$ 
33:  if  $\delta < 1.31$  then
34:    Break the outer while loop
35:  end if
36:  FES  $\leftarrow$  FES + 1
37: end while
38: end procedure

```

Step (1) *Initialize the population*. First, the MDEW algorithm initializes a population randomly, where each individual represents a potential solution $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,D}]$ ($i = 1, 2, \dots, NP$). To obtain good solutions, the MDEW algorithm emphasizes maintaining diversity in the initial population, which is formed according to the following rule:

$$x_{i,j} = L_j + rand * (U_j - L_j) \quad (12)$$

where U_j, L_j represent the upper and lower bounds, D and NP represent the dimension and population size, $rand$ denotes a random number between 0 and 1.

Step (2) *Generate the offspring*. Updating the population involves the following procedures.

(1) Introduce three mutation strategies [17], and generate the mutation vector v_i following rule:

$$v_i = \begin{cases} x_i + step & g \leq 0.3 * G \\ x_{c1} + F(x_{c1} - x_i) + F(x_{c2} - x_i) & 0.3 * G \leq g \leq 0.6 * G \\ x_{best} + F(x_{c1} - x_i) + F(x_{c2} - x_i) & 0.6 * G < g < G \end{cases} \quad (13)$$

where g and G represent the current and maximum iteration, respectively, x_{best} denotes the current best solution. Step is generated based on the Weibull distribution and the Euclidean distance of the current best solution x_{best} . Parameter F represents scaling factor. The value of F, x_{c1}, x_{c2} are computed using the following equations:

$$F = \alpha + \beta * wblrnd(0.5, 1, 1) \quad (14)$$

$$x_{c1} = \frac{x_{r1} + x_{r2}}{2} \quad (15)$$

$$x_{c2} = \frac{x_{r1} + x_{best}}{2} \quad (16)$$

where $\alpha = 0.15, \beta = 0.25, x_{r1}$ and x_{r2} are solutions randomly selected from the initial population, and $wblrnd$ denotes a random number generated from a Weibull distribution.

(2) Generate the new crossover vectors as following rule:

$$u_{i,G} = \begin{cases} v_{i,G} & \text{if } rand \leq CR \text{ or } j = n_j \\ x_{i,G} & \text{else} \end{cases} \quad (17)$$

where $u_{i,G}$ represents the new trial vector, n_j denotes a random number where $n_j \in [1, D]$, and CR indicates the crossover rate within the range of $[0, 1]$.

(3) Push the trial vector to the next generation if it has better fitness.

$$x_{i,G+1} = \begin{cases} u_{i,G} & \text{if } f(u_{i,G}) \leq f(x_{i,G}) \\ x_{i,G} & \text{else} \end{cases} \quad (18)$$

where $f(u_{i,G})$ and $f(x_{i,G})$ are the fitness values of the trial vector and target vector, respectively.

Step (3) *Updating the Kriging Model*. Offspring are selected using k-means clustering. The selected individuals are then validated through FEA to update both the dataset and the Kriging model.

Step (4) *Evaluate the population convergence threshold.* To enhance the global search capability of MDEW, a population convergence threshold is defined. the population's standard deviation in each dimension is calculated as follows:

$$\sigma_k = \sqrt{\frac{1}{NP} \sum_{i=1}^{NP} (x_{i,k} - \mu_k)^2} \tag{19}$$

$$\mu_k = \frac{1}{NP} \sum_{i=1}^{NP} x_{i,k} \tag{20}$$

where $x_{i,k}$ denotes the value of the i th dimension for the k th individual, denotes the mean value of the k th dimension.

The population convergence threshold δ is defined as the mean standard deviation across all dimensions:

$$\delta = \frac{1}{d} \sum_k \sigma_k \tag{21}$$

Through multiple convergence experiments, the convergence threshold δ in this study is set to 1.31. If $\delta < 1.31$, it indicates a high degree of population aggregation. Continuing the optimization using the MDEW algorithm at this stage would be inefficient. The Interior-Point Method (IPM), with its deterministic mathematical programming approach, enables refined local exploitation, effectively compensating for a common limitation of population-based algorithms in having limited exploitation capability and a tendency to stagnate during the final stages of the search. Otherwise, the process returns to repeat Steps 2–4.

Step (5) *Construct the approximation function.* The approximation function is constructed using the logarithmic potential function and the Lagrange multiplier as follows:

$$\begin{cases} \min_{x,s} f_\mu(x, s) = \min_{x,s} f(x) - \mu \sum_i \ln(s_i) \\ s \geq 0, \mu > 0, h(x) = 0, g(x) + s = 0 \end{cases} \tag{22}$$

The added logarithmic term acts as a barrier function, μ and S represent the barrier parameter and slack variable.

Step (6) *Update the slack variable and barrier parameter.* Each iteration applies either a direct step or a conjugate gradient step, based on an evaluation function defined as:

$$f_\mu(x, s) + v \|h(x), g(x) + s\| \tag{23}$$

(1) The direct step $(\Delta x, \Delta s)$ is defined as:

$$\begin{bmatrix} H & 0 & J_h^T & J_g^T \\ 0 & \Lambda & 0 & S \\ J_h & 0 & 0 & 0 \\ J_g & I & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} \nabla f + J_h^T y + J_g^T \lambda \\ S \lambda \\ h \\ g + s \end{bmatrix} \tag{24}$$

where J_g and J_h denote the Jacobian matrix of the constraint function g and h , $S = \text{diag}(s)$, λ represents the Lagrangian multiplier vector associated with the constraint g , e denotes unit vector, and $H = \nabla^2 f(x) + \sum_i \lambda_i \nabla^2 g_i(x)$.

(2) The conjugate gradient step obtains the Lagrangian multipliers by solving the following equation:

$$\nabla_x L = \nabla_x f(x) + \sum_i \lambda_i \nabla g_i(x) + \sum_j \lambda_j \nabla h_j(x) = 0 \quad (25)$$

When satisfying $g(x) + J_g \Delta x + \Delta s = 0$, $h(x) + J_h \Delta x = 0$, $(\Delta x, \Delta s)$ can be approximately solved using the following equation:

$$\min_{\Delta x, \Delta s} \nabla f^T \Delta x + \frac{1}{2} \Delta x^T \nabla_{xx}^2 L \Delta x + \mu e^T S^{-1} \Delta s + \frac{1}{2} \Delta s^T S^{-1} \Lambda \Delta s \quad (26)$$

Due to the black-box nature of the distributed active noise optimal control model, the finite difference method was employed to compute all first- and second-order derivatives. The validity of this method is supported by Refs. [18,19].

The parameter μ is reduced by a certain value if the solution of the approximating function from the previous iteration is meets a predefined accuracy threshold, where the accuracy is defined by:

$$\max(\|\nabla f(x) + J_g^T \lambda\|, \|S\lambda - \mu e\|, \|h\|, \|g(x) + s\|) \quad (27)$$

The barrier parameter is updated using the predictor-corrector step [20]. In the predictor step, a linearized step size is employed, and with $\mu = 0$, Eq. (24) can be updated as:

$$\begin{bmatrix} H & 0 & J_h^T & J_g^T \\ 0 & \Lambda & 0 & S \\ J_h & 0 & 0 & 0 \\ J_g & I & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} \nabla f + J_h^T y + J_g^T \lambda \\ S\lambda \\ h \\ g + s \end{bmatrix} \quad (28)$$

Define and as the maximum step sizes that satisfy the constraints.

$$\begin{aligned} \alpha_s &= \max(\alpha \in (0, 1] : s + \alpha \Delta s \geq 0) \\ \alpha_\lambda &= \max(\alpha \in (0, 1] : s + \alpha \Delta \lambda \geq 0) \end{aligned} \quad (29)$$

To evaluate the performance of the prediction-corrector step in updating the barrier parameter, a complementary parameter μ_p is introduced. It is calculated from the prediction step as follows:

$$\mu_p = \frac{(s + \alpha_s \Delta s)(\lambda + \alpha_\lambda \Delta \lambda)}{m} \quad (30)$$

where m denotes the number of constraints. The first correction step determines the search direction using the following equation:

$$\begin{bmatrix} H & 0 & J_h^T & J_g^T \\ 0 & \Lambda & 0 & S \\ J_h & 0 & 0 & 0 \\ J_g & I & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{cor} \\ \Delta s_{cor} \\ \Delta y_{cor} \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} 0 \\ \Delta S \Delta \lambda \\ 0 \\ 0 \end{bmatrix} \quad (31)$$

The second correction step, or centering step, is computed by solving the following equation:

$$\begin{bmatrix} H & 0 & J_h^T & J_g^T \\ 0 & \Lambda & 0 & S \\ J_h & 0 & 0 & 0 \\ J_g & I & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{cen} \\ \Delta s_{cen} \\ \Delta y_{cen} \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} \nabla f + J_h^T y + J_g^T \lambda \\ S\lambda - \mu e \sigma \\ h \\ g + s \end{bmatrix} \quad (32)$$

where $\sigma = (\mu_p / \mu)^3$

Step (7) *Perform iterative operations*. If the difference between the results of the last five iterations is less than 10^{-12} , repeat only step 6. Otherwise, continue repeating steps 1–6 until the maximum number of iterations (10) is reached.

3 Experimental Results

The experiments were conducted on a PC running Windows 10 Pro-64 bit, equipped with an Intel(R) Core(TM) i7-6700 CPU @ 3.40 GHz and 16 GB of RAM. The implementation of all pro-posed algorithms was done using MATLAB 9.9 (R2020b).

In the finite element simulation, the computational domain is configured as follows: each wavelength is discretized with no fewer than 8 mesh elements to ensure convergence, and a PML is applied as the absorbing boundary condition. The MULTifrontal Massively Parallel Sparse Direct Solver (MUMPS) is selected for computation, with the relative tolerance set to 0.001. The description related to the Kriging model is as follows: In the Kriging model, the kernel function employs the RBF Gaussian kernel to characterize the correlation structure of spatial data, which effectively captures the similarity between sample points through an exponential form of radial distance. The estimation of hyperparameters is based on a maximum likelihood estimation strategy, where the likelihood function is maximized to optimize model fitting, thereby enhancing prediction accuracy and robustness.

In the experimental design, the IPMDEW algorithm was employed to optimize the Kriging model. To systematically evaluate both the optimization performance of IPMDEW and the prediction accuracy of the Kriging model, multiple comparative schemes were established. For comparative analysis, both the standard DE algorithm and the MDEW algorithm—the latter of which exhibits enhanced global exploration capabilities—were introduced. During the FEA optimization process, key algorithm parameters—including population size (NP), mutation factor (F), crossover probability (CR) and number of iterations (nt)—were systematically varied and compared the variations in noise reduction performance under different parameter combinations. Based on multiple sets of experimental comparisons, the optimal parameter set was ultimately determined, with specific values provided in Table 1. All algorithms used in this study employ this parameter set for subsequent analysis.

Table 1: Parameter settings for different algorithms.

Parameter	DE	MDEW	IPMDEW
D	32	32	32
NP	100	100	100
F	0.5	–	–
CR	0.9	0.9	0.9
nt	30	30	100
δ	–	–	1.31

To assess the prediction accuracy of the Kriging surrogate model, a test sample set was generated within the parameter space using the LHS method, and the predictions of the Kriging model were compared against FEA.

The DE, MDEW, and IPMDEW algorithms were used to perform twenty independent optimization runs based on the Kriging surrogate model (Figs. 3 and 4). The convergence curves in Figs. 5 and 6, which compare the performance of the three algorithms in both Kriging-based surrogate optimization and Non-surrogate optimization, the IPMDEW algorithm exhibits significant performance advantages in optimizing both the surrogate model and the direct simulation model. These advantages are evident in its faster convergence speed and superior solution quality. Furthermore, Fig. 7 shows that the prediction accuracy of the Kriging model improved consistently throughout the iterative process, eventually stabilizing with minor fluctuations. This validates its reliability as an efficient surrogate model.

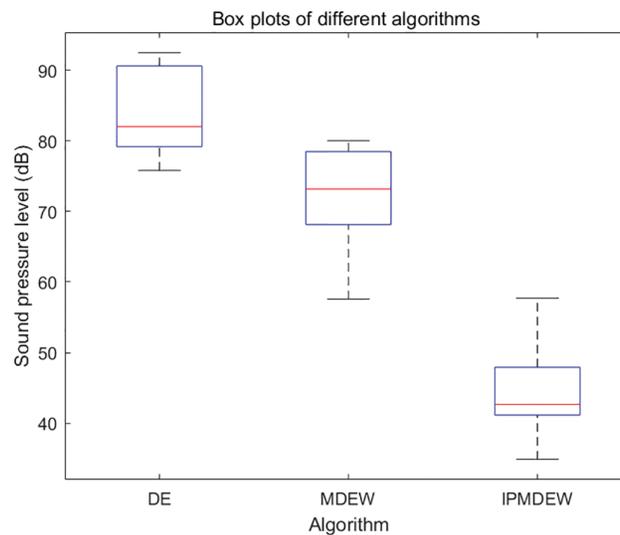


Figure 3: Box plots of multiple runs for different algorithms under surrogate optimization.

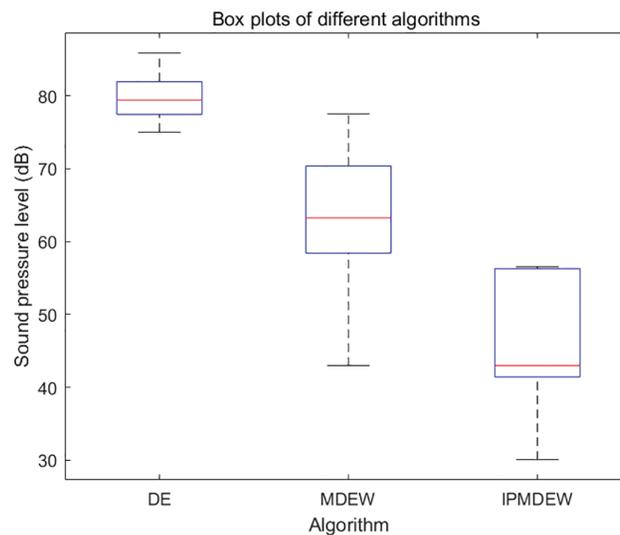


Figure 4: Box plots of multiple runs for different algorithms under Non-surrogate optimization.

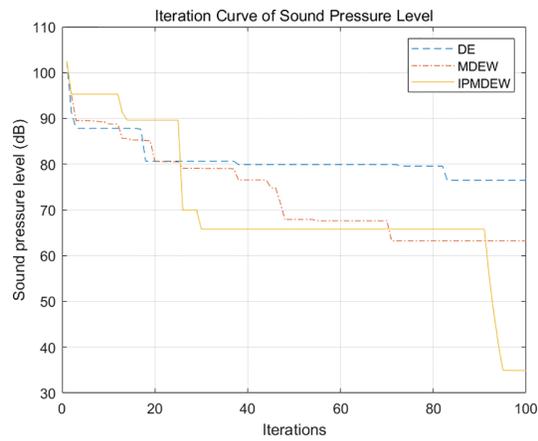


Figure 5: Iteration curves of different algorithms for Kriging surrogate optimization.

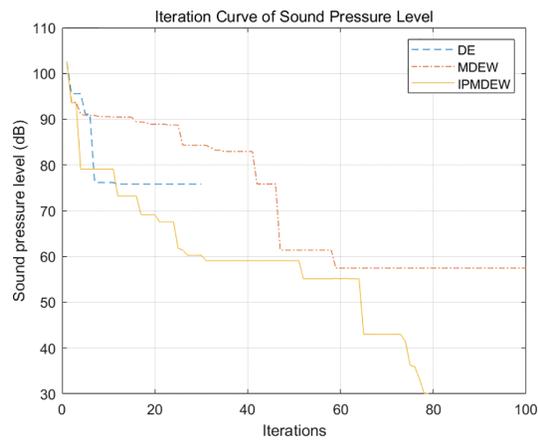


Figure 6: Iteration curves of different algorithms for Non-surrogate optimization.

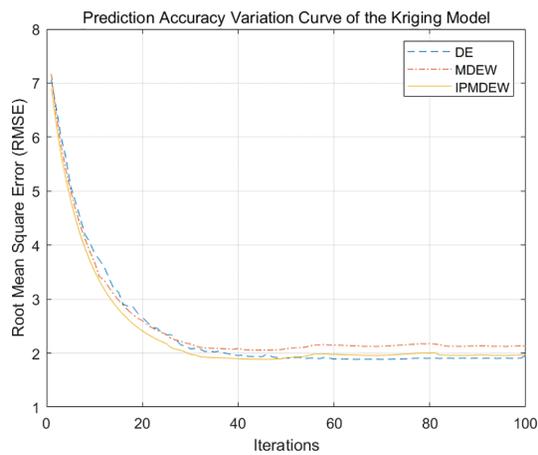


Figure 7: Prediction accuracy variation curve of the Kriging model.

A comparison of the results in [Tables 2](#) and [3](#) reveal the average performance of differences algorithms under Kriging-based surrogate optimization and Non-surrogate optimization, with the optimization outcomes further validated via finite element analysis. The DE algorithm exhibited premature convergence due to its weak global search capability. Although more effective than DE, the MDEW algorithm was limited by the Kriging model's prediction accuracy during iterations; as a result, its final noise reduction performance did not reach the optimum achieved by direct simulation. Despite these limitations, the Kriging-assisted optimization approach dramatically improved the optimization efficiency of all algorithms evaluated. The IPMDEW algorithm, however, effectively balancing global exploration and local exploitation. It achieved a noise reduction effect closely approximated the optimum while substantially improving computational efficiency, successfully co-optimizing both performance and efficiency.

Table 2: Comparison of the noise reduction performance between surrogate and non-surrogate optimization under different algorithms.

Algorithms	Surrogate	Finite Element Verification	Non-Surrogate
DE	-18.85 dB	-22.94 dB	-22.85 dB
MDEW	-29.80 dB	-32.89 dB	-38.32 dB
IPMDEW	-58.32 dB	-60.62 dB	-55.52 dB

Table 3: Comparison of optimization efficiency between Kriging-based surrogate optimization and non-surrogate optimization under different algorithms.

Algorithms	Surrogate Optimization Efficiency	Non-Surrogate Optimization Efficiency
DE	61.01 h	180.38 h
MDEW	57.45 h	404.81 h
IPMDEW	57.60 h	405.24 h

To evaluate the predictive capability of the Kriging model for optimal solutions, we validated its optimization results through finite element simulation. [Fig. 8](#) shows a comparison of the spatial sound pressure level distributions before and after optimization achieved by the three algorithms. The errors between the Kriging-assisted optimization results and the FEA results were all within 3 dB, confirming the reliability of the surrogate model.

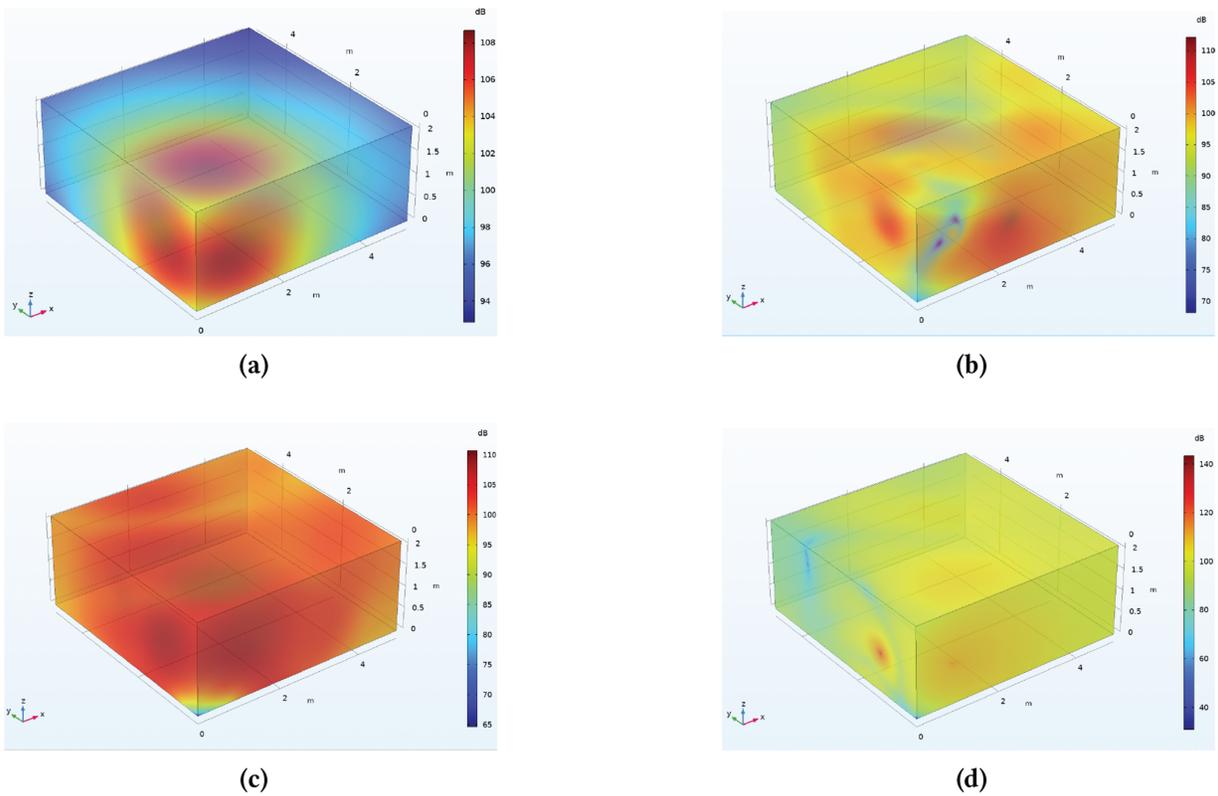


Figure 8: Finite Element validation of Kriging surrogate-assisted optimization under different algorithms. (a) baseline sound pressure level distribution; (b) DE algorithm; (c) MDEW algorithm; (d) IPMDEW algorithm.

4 Conclusion

This paper presents a surrogate-assisted hybrid differential evolution algorithm to address the high-dimensional expensive optimization problem of secondary source deployment in active noise control. By employing a Kriging model as a surrogate for finite element-based sound field simulations, combined with a k-means clustering-based dynamic update mechanism, the computational time for a single sound pressure evaluation was reduced by 85.79%, while model reliability in non-uniform sound fields was maintained. The IPMDEW algorithm integrates the global exploration capability of MDEW with the gradient-based local optimization of the IPM, achieving dynamic switching via a population diversity threshold. Compared to conventional DE and MDEW algorithms, IPMDEW improves noise reduction performance by 67.7% and 48.9%, respectively. The optimized secondary source deployment achieved a noise reduction of 58.32 dB, approaching the accuracy of finite element simulations and validating the algorithm's effectiveness and precision in complex acoustic environments. This work introduces a surrogate-based hybrid differential evolution approach to acoustic source deployment, providing a practical and efficient solution for optimizing secondary source arrangements in multi-channel spatial active noise control systems. The method has been validated for its reliability in steady sound fields; however, its adaptability to dynamic time-varying acoustic environments, such as those with real-time noise source variations, remains to be investigated. In such scenarios, the existing offline surrogate model update mechanism may struggle to cope with rapid changes, potentially leading to delays in updates and a decline in prediction accuracy.

Acknowledgement: The author would like to express the most sincere gratitude to my supervisor, for his invaluable guidance, continuous support, and patience throughout this study. His profound knowledge and rigorous scientific attitude have been a great asset to my research career.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Chao Wu: writing—original draft, software, experiment, validation. Weijian Kong: writing—review & editing, methodology, conceptualization. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data used to support the findings of this study are included in the article.

Ethics Approval: Not applicable.

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

References

1. Brancatia A, Aliabadi MH, Mallardo V. ABEM sensitivity formulation for three-dimensional active noise control. *Int J Numer Methods Eng.* 2012;90(9):1183–206. doi:10.1002/nme.3369.
2. Sahu N, Wu L, Babu P, Bhavani Shankar MR, Ottersten B. Optimal sensor placement for source localization: a unified ADMM approach. *IEEE Trans Veh Technol.* 2022;71(4):4359–72. doi:10.1109/TVT.2022.3146603.
3. Liu HQ, Ren ZG, He CL, Du W. A surrogate-assisted differential evolution algorithm with a dual-space-driven selection strategy for expensive optimization problems. *Complex Intell Syst.* 2025;11(5):230. doi:10.1007/s40747-025-01812-0.
4. Guo XQ, Wei FF, Zhang J, Chen WN. A classifier-ensemble-based surrogate-assisted evolutionary algorithm for distributed data-driven optimization. *IEEE Trans Evol Comput.* 2025;29(3):711–25. doi:10.1109/TEVC.2024.3361000.
5. Zhou RF, Ren CL, Meng ZY, Zhu HB. Surrogate-assisted differential evolution with search space tightening for high-dimensional expensive optimization problems. *IEEE Trans Syst Man Cybern Syst.* 2025;55(10):7356–68. doi:10.1109/TSMC.2025.3582897.
6. Liu G, Su X, Zhou R, Liu N, Zhang L, Tian Y. A fast generalized surrogate -assisted evolutionary algorithm for solving expensive constrained optimization problems. *J Supercomput.* 2025;81(11):1185. doi:10.1007/s11227-025-07660-6.
7. Rahi KH, Singh HK, Ray T. Partial evaluation strategies for expensive evolutionary constrained optimization. *IEEE Trans Evol Comput.* 2021;25(6):1103–17.
8. Zhang J, Li M, Yue X, Wang X, Shi M. A hierarchical surrogate assisted optimization algorithm using teaching-learning-based optimization and differential evolution for high-dimensional expensive problems. *Appl Soft Comput.* 2024;152:111212. doi:10.1016/j.asoc.2023.111212.
9. Pan JS, Liu N, Chu SC, Lai T. An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems. *Inf Sci.* 2021;561(2):304–25. doi:10.1016/j.ins.2020.11.056.
10. Jin YC. Surrogate-assisted evolutionary computation: recent advances and future challenges. *Swarm Evol Comput.* 2011;1(2):61–70. doi:10.1016/j.swevo.2011.05.001.
11. Storn R, Price K. Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim.* 1997;11(4):341–59. doi:10.1023/A:1008202821328.
12. Farda I, Thammano A, Morris J. Differential evolution with self-adaptive mutation and population improvement strategy for optimization problems. *IEEE Access.* 2024;12:131809–29. doi:10.1109/ACCESS.2024.3460385.
13. Liu X, Sun JY, Zhang QF, Wang Z, Xu Z. Learning to learn evolutionary algorithm: a learnable differential evolution. *IEEE Trans Emerg Top Comput Intell.* 2023;7(6):1605–20. doi:10.1109/TETCI.2023.3251441.

14. Chew WC, Liu QH. Using perfectly matched layers for elastodynamics. In: Proceedings of the IEEE Antennas and Propagation Society International Symposium, 1996 Digest; 1996 Jul 21–26; Baltimore, MD, USA. p. 366–9. doi:10.1109/APS.1996.549614.
15. Zhan DW, Cheng YS, Liu J. Expected improvement matrix-based infill criteria for expensive multiobjective optimization. *IEEE Trans Evol Comput.* 2017;21(6):956–75. doi:10.1109/TEVC.2017.2697503.
16. Ioardanis I, Koukouvino C, Silou I. Classification accuracy improvement using conditioned latin hypercube sampling in supervised machine learning. In: Proceedings of the 2022 12th International Conference on Dependable Systems, Services and Technologies (DESSERT); 2022 Dec 9–11; Athens, Greece. p. 1–5. doi:10.1109/DESSERT58054.2022.10018677.
17. Layeb A. Differential evolution algorithms with novel mutations, adaptive parameters, and Weibull flight operator. *Soft Comput.* 2024;28:7039–91. doi:10.1007/s00500-023-09561-3.
18. Grapiglia GN, Gonçalves MLN, Silva GN. A cubic regularization of Newton's method with finite difference Hessian approximations. *Numer Algorithms.* 2022;99(2):607–30. doi:10.1007/s11075-021-01200-y.
19. Andrei N. Diagonal approximation of the hessian by finite differences for unconstrained optimization. *J Optim Theory Appl.* 2020;185(3):859–79. doi:10.1007/s10957-020-01676-z.
20. Nocedal J, Wright SJ. Interior-point methods for nonlinear programming. In: Numerical optimization. Springer series in operations research and financial engineering. New York, NY, USA: Springer; 2006. p. 563–97. doi:10.1007/978-0-387-40065-5_19.