Kobe University Repository : Kernel

PDF issue: 2026-11-15

Gradient-Enhanced Kriging-Based Parallel Efficient Global Optimization Algorithm and Its Application in Aerodynamic Shape Optimization

Fu, Hang ; Wang, Qingyu ; Nakashima, Takuji ; Kawasaki, Asahi ; Lai, Chenguang ; Shimizu, Keigo ; Bale, Rahul ; Tsubokura, Makoto

(Citation)
IEEE Access,13:73848-73869

(Issue Date)
2025-04-15

(Resource Type)
journal article

(Version)
Version of Record

(Rights)
© 2025 The Authors.
This work is licensed under a Creative Commons Attribution 4.0 License.

(URL)
https://hdl.handle.net/20.500.14094/0100495851

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 March 2025, accepted 7 April 2025, date of publication 15 April 2025, date of current version 5 May 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3560292

RESEARCH ARTICLE

Gradient-Enhanced Kriging-Based Parallel Efficient Global Optimization Algorithm and Its Application in Aerodynamic Shape Optimization

HANG FU[1], QINGYU WANG[2], TAKUJI NAKASHIMA[3], ASAHI KAWASAKI[1], CHENGUANG LAI[4], KEIGO SHIMIZU[5], RAHUL BALE[1,6], AND MAKOTO TSUBOKURA[1,6]

[1]Graduate School of System Informatics, Kobe University, Kobe, Hyogo 6578501, Japan
[2]Institute of Auto Engineering, BYD Auto Industry Company Ltd., Shenzhen 518000, China
[3]Graduate School of Advanced Science and Engineering, Hiroshima University, Higashihiroshima, Hiroshima 739-8527, Japan
[4]College of Vehicle Engineering, Chongqing University of Technology, Chongqing 400054, China
[5]Research and Development Center, Mazda Motor Corporation, Higashihiroshima, Hiroshima 730-8670, Japan
[6]Complex Phenomena Unified Simulation Research Team, RIKEN, Center for Computational Science, Kobe, Hyogo 650-0074, Japan

Corresponding author: Makoto Tsubokura (tsubo@tiger.kobe-u.ac.jp)

This work was supported in part by the MEXT as ''Program for Promoting Researches on the Supercomputer Fugaku'' under Grant JPMXP1020210316; in part by the Computational Resources of Supercomputer Fugaku provided by the RIKEN Center for Computational Science under Project hp210262, Project hp220180, Project hp230193, and Project hp240205; and in part by China Scholarship Council (CSC) under Grant 202108500034.

ABSTRACT The parallel efficient global optimization (EGO) algorithm was developed to leverage the rapid advancements in high-performance computing. However, conventional parallel EGO algorithm based on ordinary kriging (OK) model faces limitations when applied to complex, nonlinear, and high-dimensional engineering optimization problems, as OK model relies only on sample point locations, restricting the available training information. To further enhance the optimization potential of the parallel EGO algorithm, an improved system that integrates the parallel EGO algorithm with gradient-enhanced kriging (GEK) is proposed. The integration serves two main purposes: on the one hand, GEK provides a more comprehensive representation of the design space, improving the surrogate model's predictive accuracy; on the other hand, as the number of infill points increases, GEK mitigates the unreliability of the pseudo expected improvement (PEI) function, facilitating the efficient search for high-quality global optima. To ensure high performance, various benchmark functions were used to evaluate the impact of different optimizers on the proposed system, ultimately identifying the most effective optimizer. Additionally, optimization tests were conducted on benchmark functions, airfoil aerodynamic optimization, and automotive aerodynamic optimization to compare the proposed system with traditional parallel EGO systems in optimization performance. The results demonstrate that the modified Gaussian bare-bones differential evolution (MGBDE) optimizer consistently accelerated convergence to high-quality solutions in most benchmark functions, achieving the highest average ranking across all tests. Compared to traditional parallel EGO system, the proposed method showed superior convergence speed, solution quality, and computational efficiency. Specifically, for airfoil and automotive aerodynamic optimization, when achieving the target performance improvements of 2.4% and 7.0%, the proposed system reduced the wall-clock time cost from 1.65h to 0.33h and from 203h to 14.5h, respectively, significantly improving optimization efficiency.

INDEX TERMS Gradient-enhanced kriging, efficient global optimization, parallel optimization, aerodynamic shape optimization.

The associate editor coordinating the review of this manuscript and approving it for publication was Binit Lukose.

I. INTRODUCTION

Nearly all engineering optimization problems are computationally expensive and involve complex black-box

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

functions [1]. Such problems allow for the observation of input data and output results, while the specific mathematical relationship between the two remains unclear (Figure 1). Such as finite element analysis and computational fluid dynamics. With the growing demands of modern industries for improved product quality and production efficiency, the pursuit of highly efficient optimization techniques has become a shared objective among researchers.

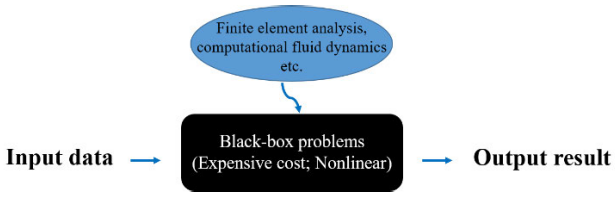


FIGURE 1. Characteristics of engineering optimization problems.

A. TRADITIONAL OPTIMIZATION

In the early stages, engineers relied solely on empirical knowledge and theoretical research to improve optimization targets, a process that required extensive and costly experimental support [2]. Advances in computing technology, especially computational fluid dynamics and computer-aided design, have greatly improved the efficiency of engineering optimization. However, the associated computational resource demands remain prohibitively high [3].

B. INTELLIGENT OPTIMIZER

With the rapid development of intelligent computing, engineering optimization is gradually overcoming its reliance on extensive human and material resources. Various optimization algorithms, including bio-inspired methods like differential evolution and genetic algorithms, as well as population-based techniques like particle swarm optimization, have significantly improved engineering optimization [4], [5], [6]. However, traditional optimization algorithms are prone to becoming trapped in local optima. To address this issue, recent years have witnessed the emergence of enhanced and novel optimization algorithms [7], [8], [9], [10], [11], [12]. Such advanced optimizers have effectively mitigated the limitations of traditional algorithms in avoiding entrapment in local optima. However, applying these algorithms directly often requires thousands or even tens of thousands of objective function evaluations [13], making them impractical for product design.

C. SURROGATE-BASED OPTIMIZATION (SBO)

To overcome the high computational cost and inefficiency associated with directly applying optimizers, surrogate-based optimization (SBO) methods have attracted significant attention in recent years [14]. Figure 2 shows the typical process of SBO in vehicle shape optimization: sampling, deformation, experimentation, surrogate model construction, and search optimal solution. SBO provides a cost-effective way to predict function values and is widely used in engineering applications [15], such as gas turbine blade shape optimization [16], airfoil optimization [17], [18], helicopter rotor blade design [19], automotive aerodynamic performance optimization [20], [21], aerodynamic optimization of streamlined bridge deck [22] and design optimization for battery thermal management system [23].

D. ARTIFICIAL INTELLIGENCE

In recent years, emerging research hotspots in the field of engineering optimization have increasingly focused on artificial intelligence technologies, such as deep neural networks and deep reinforcement learning [24], [25], [26], [27], [28]. Current research indicates that high predictive accuracy is achieved in forecasting objective function values. However, high fidelity in training data, substantial computational resource demands, and limited model interpretability present significant challenges, which constrain the applicability of such methods in practical engineering optimization problems. Surrogate model-based optimization methods, by contrast, offer a more mature and reliable approach for practical engineering optimization applications.

E. KRIGING-BASED OPTIMIZATION

Among traditional surrogate models, kriging is widely used in engineering for its ability to predict unknown points and quantify uncertainty [29], [30]. However, the accuracy of the ordinary kriging (OK) model, constructed with a limited number of samples, may not meet the requirements of practical engineering optimization problems. The OK model, as an approximation of the true function, deviates in regions with sparse samples. Limited samples also constrain the model's training, leading to suboptimal solutions and a higher risk of missing the global optimum.

Two key approaches address the existing limitations. On the one hand, Forrester et al. [31] proposed extended regression kriging with re-interpolation, which automatically filters noisy data introduced by experiments and yields a higher-fidelity surrogate model. To extract more efficient information from the existing sample set, Laurenceau et al. [32] used function values at sample points to build a surrogate model and incorporated additional information, such as gradient values, to aid the construction process. The collaborative approach has demonstrated effectiveness in practical engineering optimization problems [33], [34], [35], [36], [37]. Additionally, Han et al. [38], [39] proposed kriging-based variable-fidelity surrogate models to improve the efficiency and accuracy of kriging-based optimization method. However, relying solely on this approach often reaches the fitting limit of the sample points. This makes it challenging to address highly nonlinear and high-dimensional optimization problems.

On the other hand, kriging is well-suited for developing a point infill criterion, as it predicts unknown points and quantifies the uncertainty of those predictions. Building on

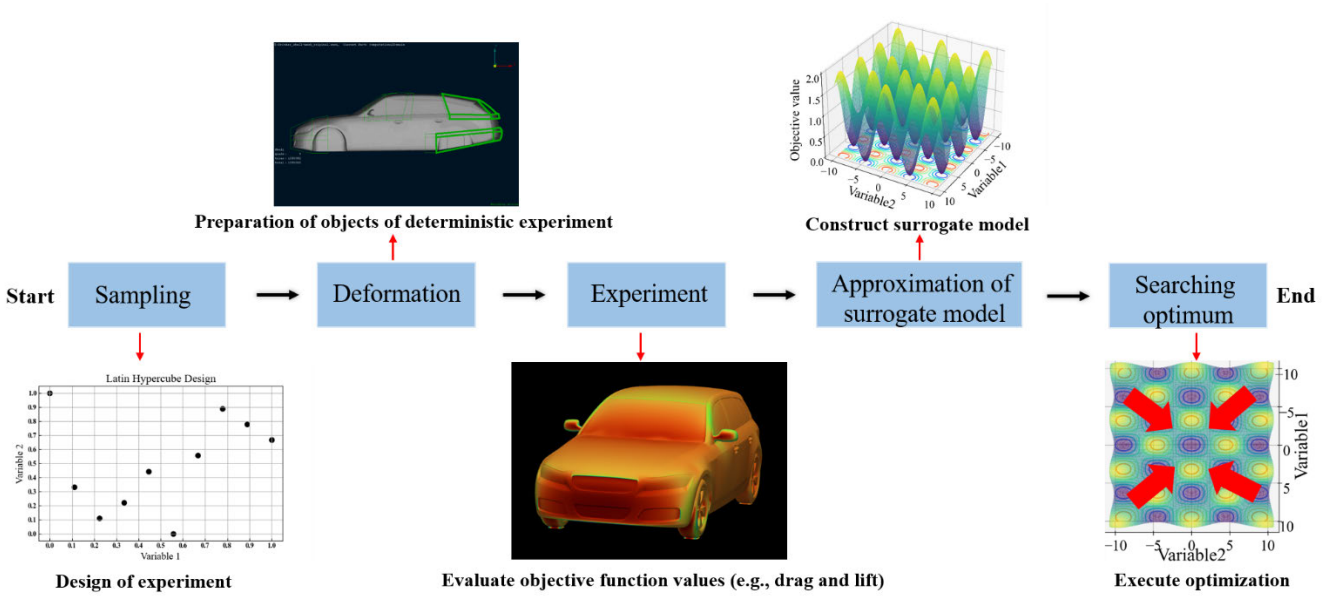


FIGURE 2. The typical process of surrogate-based optimization method (e.g. vehicle shape optimization).

this concept, Jones et al. [40] proposed the efficient global optimization (EGO) algorithm, which utilizes the expected improvement (EI) criterion to identify new sample points with the highest potential for improvement. This approach achieves better efficiency and accuracy than previous methods and has been widely applied in practical engineering optimization problems [41], [42]. However, during each point infill cycle, only one potentially improving sample point is added to the training sample set. Consequently, external computational resources are not fully utilized, limiting the improvement in optimization efficiency. To address this issue, parallel point infill criterions have been developed. Initially, Ginsbourger et al. [43] proposed the q-point expected improvement (q-EI) criterion, which selects multiple points based on a criterion in each cycle. Despite enhancing optimization efficiency, the method remains time-consuming. Further improvements include the kriging believer (KB) and constant liar (CL) strategies. The KB criterion relies on the analytically known EI and optimization in d dimensions, which is computationally affordable. However, over-relying on the predictor outside the observed data range may result in convergence to suboptimal regions. The CL strategy considers the current minimum value to be a ''liar'' and updates the kriging model to give the next candidate point. Although the computational cost of the CL strategy is much lower than that of the q-EI criterion, reliance on fabricated information from the 'liar' leads to exploration in poorer regions, causing the optimization to become trapped in local optima. In recent years, Zhan et al. [44] proposed a parallel point infill criterion, called pseudo expected improvement (PEI), based on the observed behavior of the EI function. The approach simulates the changes in the EI function caused by adding a new sample point to the training set, without updating the kriging model. As a result, the parallel EGO algorithm with the PEI criterion demonstrates superior performance and efficiency compared to other parallel EGO algorithms. Zhan et al. proposed a parallel EGO algorithm based on the OK surrogate model. However, its optimization efficiency can be further improved by enhancing the accuracy of the surrogate model. The PEI function is constructed by multiplying the original EI function with multiple influence functions (IFs), corresponding to the number of parallel sample points. As the number of parallel sample points increases, the reliability of the PEI function declines significantly due to the limited accuracy of the OK model with a finite number of training samples, especially in high-dimensional optimization problems. Increasing the number of training samples can improve model accuracy but may lead to overfitting [45]. Wang et al. [46] attempted to enhance optimization efficiency by integrating the regression kriging model with parallel EGO. However, since regression kriging primarily smooths the fitting trend by filtering numerical noise, its contribution to improving optimization efficiency remains limited. Existing parallel EGO studies mainly use only the spatial positions of training samples while ignoring gradient information, which reflects changes in spatial position. Therefore, integrating gradient-enhanced kriging (GEK) with parallel EGO may significantly improve optimization efficiency and enable the handling of high-dimensional engineering optimization problems, making it a promising direction for further investigation.

To further enhance the optimization potential and efficiency of the parallel EGO algorithm and achieve the greatest possible improvements within a reasonable computational cost for engineering optimization problems, this paper proposed an improved GEK-based parallel EGO algorithm

system. This system integrates gradient-enhanced techniques, a parallel point infill criterion, and powerful optimizer engines. The main contributions of this work are as follows:

a) Development of a parallel EGO system based on the GEK surrogate model using Python and an object-oriented programming approach.
b) Investigation of the impact of different advanced optimizers on the system's performance to identify a highly robust optimizer engine.
c) Optimization tests on multiple benchmark functions to compare the proposed system with the previous version in terms of convergence speed, optimization efficiency, and computational cost.
d) Optimization tests on two engineering case studies to evaluate performance differences between the proposed and previous systems in terms of convergence speed, optimization efficiency, and computational cost.

The organization of the paper is as follows: Section II problem statement and solution; Section III presents a brief introduction to the theoretical approach; Section IV proposes the improved optimization system; Section V focuses on the results and discussion, while Section VI concludes the study and outlines potential directions for future research.

II. PROBLEM STATEMENT AND SOLUTION

A. PROBLEM STATEMENT

The introduction provides a detailed analysis of the challenges encountered in practical engineering optimization problems and outlines the corresponding solutions proposed in existing research. The focus of this study is on the parallel EGO global optimization algorithm based on the kriging surrogate model. While the parallel point-filling criterion suits the computational capabilities of modern supercomputers, the OK surrogate model struggles to construct a highly reliable surrogate with limited training samples, especially for high-dimensional, highly nonlinear engineering optimization problems. Insufficient representation of critical information in the true design space can hinder the identification of high-quality approximate optimal solutions during the point-filling process, reducing the efficiency of parallel optimization.

B. PROPOSED SOLUTION

To overcome the limitations of the current parallel EGO algorithm, a combined approach integrating the GEK surrogate model with the parallel EGO algorithm is proposed. The GEK surrogate model improves performance in high-dimensional, highly nonlinear optimization problems by effectively using gradient information from the samples to enhance fitting accuracy. This enhancement improves the representation of critical information in the design space, increasing the likelihood of identifying high-quality approximate optimal solutions and boosting the efficiency of parallel optimization.

III. MATHEMATICAL FRAMEWORK

A. GRADIENT-ENHANCED KRIGING

When building a kriging model, it is crucial to use design variables and objective function values effectively while incorporating gradient information to enhance surrogate model construction. GEK model is an extension of OK model that assumes a set of p initial training samples, each with n dimensions. The objective value at an unknown point is predicted as follows:

$$\hat{y}\left(x_{p+1}\right) = \hat{\mu} + \boldsymbol{r}^T \boldsymbol{R}^{-1}\left(\boldsymbol{y} - \boldsymbol{F}\hat{\mu}\right) \tag{1}$$

where $\hat{\mu}$ represents the mean term, which is determined by solving the following weighted least-squares problem:

$$\hat{\mu} = \left(\boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{F}\right)^{-1} \boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{y} \tag{2}$$

in which $\boldsymbol{F}$ is the regression matrix representing the zeroth-order basis functions. In GEK model, the polynomial vector of the basis function is expanded to include the derivatives of the regression basis functions. In the case of zeroth-order basis functions, a vector consist of 1s and 0s is given by:

$$\boldsymbol{F} = [\mathbf{1}\ \mathbf{0}] = \left[\underbrace{1, \cdots, 1}_{p}, \underbrace{0, \cdots, 0}_{p \times n}\right]^T \tag{3}$$

The correlation vector $\boldsymbol{r}$ represents the correlation between the sample points and the unknown prediction point, whereas the correlation matrix $\boldsymbol{R}$ represents the correlation between the sample points. As GEK model incorporates gradient information, the corresponding correlation vector and correlation matrix include gradient terms:

$$\boldsymbol{r} = \begin{bmatrix} \boldsymbol{q}^1 \\ \boldsymbol{q}^2 \end{bmatrix} \tag{4}$$

Here, $\boldsymbol{q}^1$ represents the correlation vector used in the OK model and $\boldsymbol{q}^2$ contains the first-order derivatives of $\boldsymbol{r}$:

$$\begin{cases} \boldsymbol{q^1} = \begin{bmatrix} \psi\left(x^{n+1}, x^1\right) \\ \vdots \\ \psi\left(x^{n+1}, x^p\right) \end{bmatrix} \\ \boldsymbol{q^2} = \begin{bmatrix} \frac{\partial \psi\left(x^{n+1}, x^1\right)}{\partial x^1} \\ \vdots \\ \frac{\partial \psi\left(x^{n+1}, x^p\right)}{\partial x^p} \end{bmatrix} \end{cases} \tag{5}$$

The Gaussian correlation function is selected:

$$\psi\left(x^i, x^j\right) = exp\left[\sum_{k=1}^{n} -\theta_k \left(\left\|x_k^i - x_k^j\right\|\right)^2\right] \tag{6}$$

where θ_k represents the hyper-parameter that controls the correlation variation along the kth dimension. It determines how the correlation between sample points changes with distance. As shown in Figure 3, the horizontal axis represents the distance between two sample points, while the vertical axis represents their correlation. A larger hyper-parameter value

causes a more rapid change in correlation as the distance increases. The determination of hyper-parameters involves using the maximum likelihood estimation method to identify the optimal distribution trend for the given training samples [47]. Similarly, the correlation matrix is extended to include gradient terms:

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{Q}^{1,1} & \boldsymbol{Q}^{1,2} \\ (\boldsymbol{Q}^{1,2})^T & \boldsymbol{Q}^{2,2} \end{bmatrix} \tag{7}$$

Here, $\boldsymbol{Q}^{1,1}$ represents the correlation matrix used in OK:

$$\boldsymbol{Q}^{1,1} = \begin{bmatrix} \psi\left(x^1, x^1\right) & \cdots & \psi\left(x^1, x^p\right) \\ \vdots & \ddots & \vdots \\ \psi\left(x^p, x^1\right) & \cdots & \psi\left(x^p, x^p\right) \end{bmatrix} \tag{8}$$

$\boldsymbol{Q}^{1,2}$ contains the first-order derivatives of $\boldsymbol{R}$:

$$\boldsymbol{Q}^{1,2} = \begin{bmatrix} \frac{\partial \psi\left(x^1, x^1\right)}{\partial x^1} & \cdots & \frac{\partial \psi\left(x^1, x^p\right)}{\partial x^p} \\ \vdots & \ddots & \vdots \\ \frac{\partial \psi\left(x^p, x^1\right)}{\partial x^1} & \cdots & \frac{\partial \psi\left(x^p, x^p\right)}{\partial x^p} \end{bmatrix} \tag{9}$$

and $\boldsymbol{Q}^{2,2}$ contains the second-order derivatives of $\boldsymbol{R}$:

$$\boldsymbol{Q}^{2,2} = \begin{bmatrix} \frac{\partial^2 \psi\left(x^1, x^1\right)}{\partial x^1 \partial x^1} & \cdots & \frac{\partial^2 \psi\left(x^1, x^p\right)}{\partial x^1 \partial x^p} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \psi\left(x^p, x^1\right)}{\partial x^p \partial x^1} & \cdots & \frac{\partial^2 \psi\left(x^p, x^p\right)}{\partial x^p \partial x^p} \end{bmatrix} \tag{10}$$

The correlation vector and matrix clearly show that GEK model incorporates gradient information, expanding their dimensions from p × 1and p × p to p × (p + 1) and p × (p + 1) × p × (p + 1), respectively. This expansion allows GEK model to better approximate steep or highly nonlinear regions and extract more useful information from limited training samples, leading to a more accurate representation of the design space and improved prediction accuracy. However, the increased correlation vector and matrix size also raise computational costs. For low-dimensional optimization problems, the cost of GEK model fitting is negligible compared to the function evaluation cost in actual optimization, giving it a clear advantage over OK model. As dimensionality increases, this advantage diminishes. The specific cost-growth trend with increasing dimensions requires further experimental analysis.

In OK model, the vector $\boldsymbol{y}$ contains the objective function values of all initial sample points, while in GEK model, the vector also includes the derivatives of the objective function with respect to the design variables:

$$\boldsymbol{y} = \begin{bmatrix} f\left(x^1\right) \\ \vdots \\ f\left(x^p\right) \\ \frac{\partial f\left(x^1\right)}{\partial x^1} \\ \vdots \\ \frac{\partial f\left(x^p\right)}{\partial x^p} \end{bmatrix} \tag{11}$$

The mean square error of the GEK model prediction can be expressed as:

$$s^2\left(x^{p+1}\right) = \hat{\sigma}^2 \left[1 - \boldsymbol{r}^T \boldsymbol{R}^{-1} \boldsymbol{r} + \frac{\left(1 - \boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{r}\right)^2}{\boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{F}}\right] \tag{12}$$

The variance of the prediction system is given by:

$$\hat{\sigma}^2 = \frac{1}{n\,(1+p)} (\boldsymbol{y} - \boldsymbol{F}\mu)^T \boldsymbol{R}^{-1} (\boldsymbol{y} - \boldsymbol{F}\mu) \tag{13}$$

For detailed derivations regarding GEK model prediction, see Ollar et al. [47].

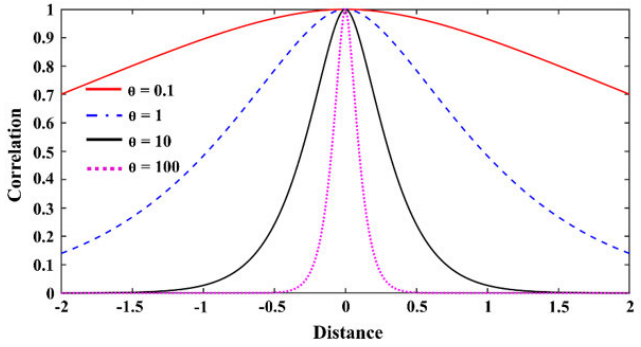


FIGURE 3. The trend of correlation changes between sample points under different hyper-parameter values (one-dimensional correlation function).

B. PARALLEL POINT INFILL CRITERION

The kriging surrogate model provides predictions of the objective function values at unknown points and offers estimates of prediction uncertainty. Thus, it is appropriate to complement the kriging surrogate model with a point infill criterion. The process of predicting the objective function values at unknown points using kriging can be regarded as implementing a stochastic process that conforms to a normal distribution with a mean of $\hat{y}(x)$ and a variance of $s^2(x)$. Hence, a point infill criterion that assesses the potential improvement at unknown points is expressed as [40]:

$$\begin{cases} z\,(x) = \dfrac{y_{min} - \hat{y}\,(x)}{s\,(x)} \\ EI\,(x) = \left(y_{min} - \hat{y}\,(x)\right) \Phi\,(z\,(x)) + s\,(x)\,\phi\,(z\,(x)) \end{cases} \tag{14}$$

where $\Phi(x)$and $\phi(x)$ represent the standard normal cumulative distribution function and probability density function, respectively. The EI function consists of two components: the first component exploits local optima when a point's predicted function value is lower than the current best, while the second promotes exploration of the design space when the prediction error is high. Therefore, the EI function itself maintains a balance between local exploitation and global exploration. Additionally, the EI function is equal to zero at the sample points, ensuring that sample points cannot be selected as potential improvement points. However, for traditional EGO algorithms, the optimization process does

not fully leverage the external computational resources to reduce the development time, because only one new point can be added to the training set in each iteration. To overcome this limitation, a novel parallel point infill PEI criterion has been proposed, allowing the identification of multiple potential improvement points in a single cycle by multiplying multiple influence functions (IFs) with the original EI function. The approach eliminates the need to evaluate the function value of an update point before updating the EI function. The PEI criterion is computed using the following formula [44]:

$$\begin{cases} IFs\left(x, q-1\right) = IF\left(x, x^{p+1}\right) \times \cdots \times IF\left(x, x^{p+q-1}\right) \\ PEI\left(x, q-1\right) = EI\left(x\right) \times IFs\left(x, q-1\right) \end{cases} \tag{15}$$

where q represents the number of potential improvement points added to the training sample set in each point infilling cycle. Based on the PEI criterion, when adding q updating points to the training set in each cycle, the first point is selected by maximizing the original EI function, while the remaining points are determined by maximizing the PEI function. The IFs are computed as follows:

$$IF\left(x, x^{e}\right) = 1 - Corr\left[Y\left(x\right), Y\left(x^{e}\right)\right] \tag{16}$$

where x^e represents the location of the updating point and $Corr\left[\cdot\right]$ denotes the correlation function, defined in the same way as in Eq.(6). According to the properties of the IFs, the PEI function equals zero at the updating points and approaches a value of one in regions distant from the updating points. Such a characteristic ensures that, after adding an updating point, the PEI criterion maintains a trend similar to that of the EI criterion, eliminating the need to update the kriging model.

IV. IMPROVED OPTIMIZATION SYSTEM FOR OPTIMIZATION OF BLACK-BOX PROBLEMS

A. GRADIENT-ENHANCED KRIGING BASED PARALLEL EGO SYSTEM

To address complex engineering optimization problems and further improve optimization efficiency through making full use of external computational resources, an improved system, namely, GEK based parallel EGO, is proposed. Specifically, to explain the idea, a relatively high approximation quality of surrogate model can be easily obtained through the effective utilization of additional gradient information. In addition, the updating points identified by the PEI criterion provide more effective gradient information for surrogate model training, which helps reduce the total number of cycles required for inserting points. The methodology enables parallel implementations, facilitating the gradual updating of the surrogate model and guiding the optimization process toward true global convergence.

Figure 4 shows the framework of GEK based parallel EGO algorithm system. The algorithmic system primarily consists of two components: initializing training samples and performing optimization cycles. For initialization, an appropriate design of experiment is used to generate initial sample points, which are then evaluated through deterministic experiments to obtain objective function values. The resulting sample set is used for surrogate model training. During the optimization process, an external optimizer is used to optimize the hyper-parameters of the surrogate model to achieve the best possible fit. The external optimizer is then employed to optimize the parallel point-filling criterion (PEI function) closely related to the current surrogate model, identifying the locations of multiple updating sample points. The objective function values at the updating sample points are re-evaluated. The new sample set is incorporated into the existing sample set, forming a new training sample set. The process continues until the desired optimal solution is found. Algorithm 1 presents the pseudocode for the parallel EGO algorithm.

As the algorithmic system involves the optimization of hyper-parameters for the kriging surrogate model and the parallel point-filling criterion (PEI function) for the updating sample points, the optimizer serves as a critical search engine in this process. Therefore, the optimization performance of the optimizer directly impacts the optimization performance of the algorithmic system. The following sections will discuss how the optimizer impacts the algorithmic system's performance, examine the effects of different optimizers, and identify the most robust optimizer for the system.

Algorithm 1 The kriging-Based Parallel EGO

Input: X (initial sample points), f (objective function), S (kriging model), q (number of parallel infills), C (stopping criterion)
Output: x_opt (optimized solution)

```
D ← InitializeDataset(X, f)
S ← TrainSurrogateModel(D)
while not C do
PEI ← ConstructPEI(S)
X_new ← OptimizePEI(PEI, q)
Y_new ← EvaluateFunction(f, X_new)
D ← UpdateDataset(D, X_new, Y_new)
S ← RetrainSurrogateModel(D)
end while
x_opt ← SelectBestSolution(D)
```

B. KEY ROLE OF OPTIMIZER IN SURROGATE BASED PARALLEL EGO SYSTEM

Based on the theory described above, if $q$potential improvement points are to be selected in each point infill cycle, then $m \times (q+1)$ optimization processes are sequentially performed for the GEK-based parallel EGO algorithm, where m represents the total number of cycles. Zhan et al. [44] found that, in large-scale and high-dimensional optimization problems, the resulting PEI function often exhibits complex and highly multimodal properties, making it very difficult for optimizers to identify globally optimal solutions on the PEI function. In the parallel EGO algorithm, during each point infill cycle, the first updating point is located using the original EI function. The remaining updating points are

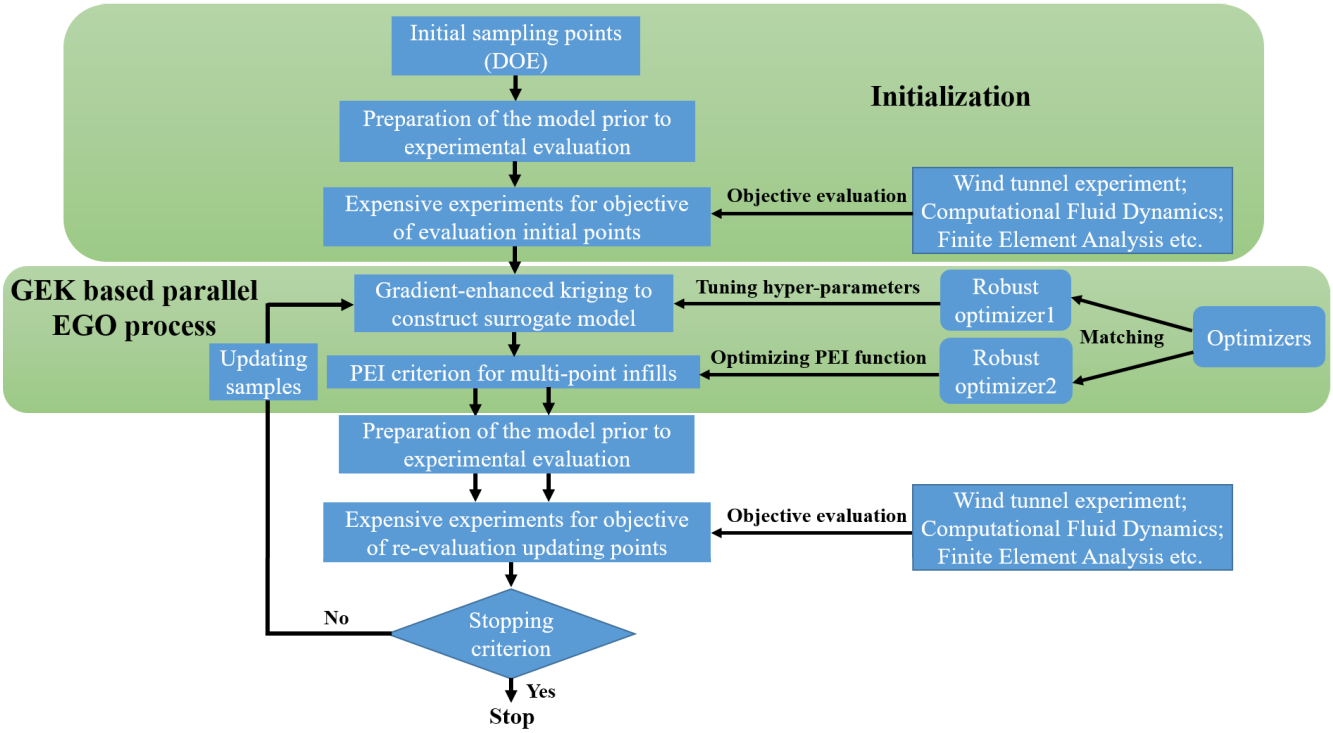


FIGURE 4. Framework of gradient-enhanced Kriging based parallel EGO algorithm system.

then determined by multiplying the corresponding IFs with the original EI function.

The quality of the earlier updating points indirectly impacts the quality of the later updating points throughout the entire optimization process, ultimately affecting the quality of the solution. Zhan et al. [44] briefly studied the influence of the DE algorithm with different parameter settings on the performance of the OK-based parallel EGO algorithm. The results revealed that the performance of the optimizer has a significant impact on the convergence performance of the OK-based parallel EGO algorithm. Given the relatively low computational cost of evaluating the PEI function, it was also recommended to conduct as many evaluations as possible to ensure the global maximum of the PEI function is located. However, the investigation only explored the influence of the DE algorithm with different parameter settings on the convergence performance of the OK-based parallel EGO algorithm and the tested benchmark functions were relatively simple. Therefore, Wang et al. [46] systematically studied the influence of various optimizers and benchmark functions on the performance of OK based parallel EGO algorithm. The results demonstrated that optimizers with a higher central goal of exploration–exploitation promote faster convergence to high-quality solutions. Which further confirmed that the performance of the optimizer significantly influences the performance of OK-based parallel EGO algorithm. However, it is clearly see that the tuning hyper-parameters [47], [48] and the optimizing PEI function are two different optimization processes, thereby the optimization strategies may differ. Previous research using the same optimizer to tune the hyper-parameters and optimize the PEI function treated the two optimization processes as a unified complete. Which will bring challenge for analyzing the specific mechanisms whereby various optimizers influences the convergence performance of the parallel EGO algorithm. Therefore, it is necessary to separate two optimization processes, keeping one optimizer fixed for one process while using different optimizers for the other process. Furthermore, as the GEK model incorporates into optimization system, its correlation vectors and matrices become more complex, leading to a significant increase in the difficulty of hyper-parameter tuning and PEI function optimizing. It is possible that previous conclusions are no longer applicable to the proposed new system.

To further explain the positive impact of the differences between EI and PEI on the convergence of the parallel EGO algorithm, the ordinary EGO and parallel EGO algorithms were used to perform point infills on the one-dimensional Griewank function within the feasible region $[-10, 10]$. The function has a global minimum of zero at point 0. The parallel EGO algorithm inserts two points in each cycle.

Figure 5 illustrates the differences in the EI and PEI functions during the point infill process. Figure 5a shows the respective locations of the second point infills obtained by the ordinary EGO and parallel EGO algorithms. After updating the first point, certain differences arise between the EI and PEI functions due to the ordinary EGO algorithm

updating the new EI function through the updated kriging model. In contrast, the parallel EGO algorithm does not update kriging model, but instead simulates the behavior changes of the EI function using the IFs. Therefore, the PEI function approximates the current EI function. Both methods select second point infills located a certain distance from the optimal value of the test function. Figure 5b shows the respective locations of the third point infills obtained by the ordinary EGO and parallel EGO algorithms. Evidently, the EGO algorithm with the PEI criterion finds a point infill location that is closer to the optimal solution than the point infill selected by the EGO algorithm with the EI criterion. Therefore, a reasonable difference between the PEI and EI functions facilitates faster convergence to high-quality solutions for the parallel EGO algorithm. It is worth noting that the performance of the optimizer in finding the maximum potential improvement points in the PEI function affects the approximation quality of the EI and PEI functions. Typically, an optimizer with higher global search capabilities ensures a better approximation of the PEI and EI functions. Conversely, an optimizer with insufficient global search capabilities may amplify the differences between the PEI and EI functions. Hence, it is worth exploring whether this elegant balance of approximation and differences between EI and PEI induced by the IFs and optimizer will be disrupted when the GEK model incorporates into the system.

C. DETERMINATION OF CANDIDATE OPTIMIZERS

To explore the influence of different optimizers on convergence performance of GEK based parallel EGO system. The gradient-based GS-SQP optimizer is fixed to tune the hyperparameters [47]. Different optimizers were selected to search for potential improvement points on the PEI function. Experiments were conducted on seven benchmark test functions (described in Section V). The final results determine the most suitable optimizer for the proposed system and provide a reference for other investigators and developers to select appropriate optimizers.

The candidate optimizers include two traditional algorithms, DE [49] and PSO [50]; two recently enhanced DE-based algorithms, enhanced adaptive guided DE (EAGDE) [8] and chaotic local search-based JADE (CJADE) [9]; a hybrid optimizer, modified Gaussian bare-bones DE (MGBDE) [10], which demonstrated the best performance in Wang et al. [46]; and a novel algorithm inspired by the hunting behavior of gorilla groups, the artificial gorilla troop optimizer (GTO) [11]. The optimizers were selected for their distinct advantages, allowing an analysis of the influence of different aspects on the performance of the proposed system. DE, EAGDE, and CJADE focus on global exploration, but exhibit relatively slow convergence speeds. PSO accelerates the entire optimization process through individual learning and social cooperation. MGBDE emphasizes the balance between local exploitation and global exploration by combining the advantages of PSO and DE. The nature-inspired

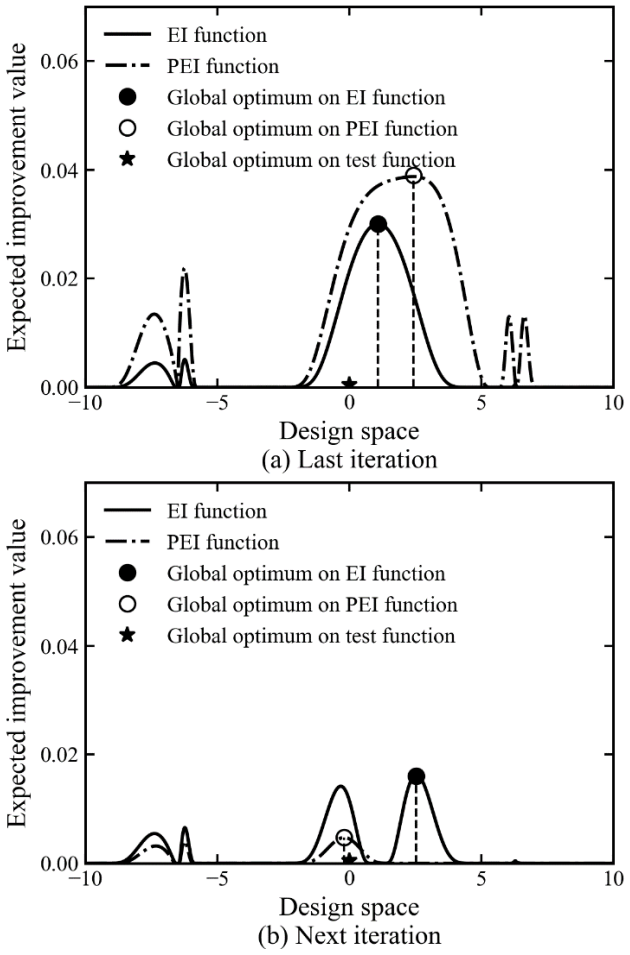


FIGURE 5. Comparison of changes in EI and PEI functions during the process of point infills.

GTO enhances the global search capability by using various random mechanisms. In summary, the impact of the optimizers on the performance of GEK based parallel EGO system is investigated in terms of the convergence speed, balance between local exploitation and global exploration, and global search capability.

V. RESULTS AND DISCUSSION

A. MATCHING THE OPTIMAL OPTIMIZER FOR THE PROPOSED ALGORITHM SYSTEM

To match the most suitable optimizer for the GEK based parallel EGO algorithm system, this section performs a comprehensive investigation into the influence of different types of optimizers on the performance of the proposed system.

1) EXPERIMENTAL SETTING

As discussed in Chapter IV-C, the candidate's optimizers have been identified, and the rational for the selection has been thoroughly explained. To guarantee a comprehensive evaluation of each optimizer's impact on the performance of the proposed system, the parameters setting for each

TABLE 1. Parameters setting for each optimizer.

Optimizer	Parameters	Value
GTO	P; β;w	0.03; 3; 0.8
PSO	c1; c2;w	1.49618; 1.49618; 0.7298
DE	F; cr	0.8; 0.9
MGBDE	F; Mu_cr; Sigma_cr	0.5; 0.5; 0.1
EAGDE	F; p_rate; N_{min}	Rand(0.1, 0.9); 0.1; 12
CJADE	Mu_F; Mu_cr; ρ	0.5; 0.5; 0.988

TABLE 2. Performance comparison of different optimizers.

Ranking	Global search capability	Complexity	Convergence speed
1	CJADE	PSO	PSO
2	EAGDE	DE	MGBDE
3	MGBDE	MGBDE	EAGDE
4	GTO	EAGDE	CJADE
5	DE	CJADE	DE
6	PSO	GTO	GTO

optimizer were set to the optimal working states, determined through numerous tests. Table 1 provides the detailed parameter settings for each optimizer. A consistent population size of 50 and a maximum iteration count of 500 were applied across all optimizers. The tests focused on optimizing benchmark functions. Please refer to the next section for the selection of benchmark functions and parameter setting of the algorithm system. Additionally, before analyzing the impact of the different optimizers on the performance of the proposed system, a comparison was conducted to better understand the inherent performance of each optimizer. Three key performance metrics were evaluated: global search capability, algorithm complexity, and convergence speed through a series of algorithm tests. Table 2 presents the comparison results.

2) RESULTS AND DISCUSSION OF COMPARATIVE INVESTIGATIONS

To clearly illustrate the impact of different optimizers on the performance of the proposed system, the results obtained from running the algorithm system with six distinct optimizers are compared. Figure 6 shows the results of the Wilcoxon signed-rank test, comparing the convergence levels of MGBDE, PSO, and DE against other optimizers across all tested functions, respectively. The symbol "=" indicates that there is no significant difference in performance between the two compared optimizers on the proposed system. Furthermore, although p-values between 0.05 and 0.10 for the significance level cannot reflect significant differences in the performance of the two compared optimizers on the proposed system, the symbols "≥" and "≤" are still included in the figure to show that the first optimizer potentially outperforms or underperforms the second optimizer, respectively, on the proposed system. It is clearly observed that there are no significant differences in performance between the PSO, DE, and MGBDE optimizers for the most of tested functions. For the *f10* and *FMSW*functions, PSO achieves significantly better performance than MGBDE and DE, respectively. Whereas for the function *f12*, MGBDE significantly outperforms DE. Although most of the statistical results for PSO, DE, and MGBDE in Figure 6 are marked with the "=" symbol, indicating comparable performance, MGBDE demonstrates a slightly better performance than the other two optimizers. Evidence for this can be found in Figure 6a, where most of the bar charts for MGBDE are positioned above the x-axis. Furthermore, from an overall perspective, PSO, DE, and MGBDE demonstrate significantly better performance than EAGDE, CJADE, and GTO. The statistical results support this observation, as PSO, DE, and MGBDE are more frequently assigned the symbols +, ≥, and = in comparisons with EAGDE, CJADE, and GTO, respectively. Consequently, the results suggest that MGBDE, DE and PSO improve the convergence performance of the GEK based parallel EGO algorithm system within a limited number of samples.

The rankings of each optimizer for each test function are more intuitively presented in Figure 7. MGBDE achieves the top ranking for test functions f11, f12, f14, and f14-l, and also performs well on f22 and FMSW, securing second place. Performance shows inferiority only on f10, lagging behind most of the other optimizers. The traditional PSO, known for its fast convergence speed, exhibits performance similar to that of MGBDE. In Figure 7, PSO demonstrates dominant performance on functions, and its ranking results indicate that this optimizer enables the proposed system to converge to high-quality solutions for most of the other test functions. Furthermore, the traditional DE, known for its strong global search capability, shows slightly inferior performance compared to MGBDE and PSO. However, DE outperforms the other three optimizers, as evidenced by the

mean ranking results in Table 3. Surprisingly, the two DE variants, EAGDE and CJADE, which are known for their stronger global search capabilities than DE, do not provide any significant advantages in improving the convergence performance of the proposed system. This conclusion is further supported by the mean rankings in Table 3, where both EAGDE and CJADE are ranked fourth. The recently developed GTO performs reasonably well only on function f14, but its performance is among the weakest across the remaining test functions.

In summary, the hybrid optimizer MGBDE, which with a high central goal of exploration–exploitation, achieved superior convergence performance in the GEK based parallel EGO algorithm system by balancing the approximation and

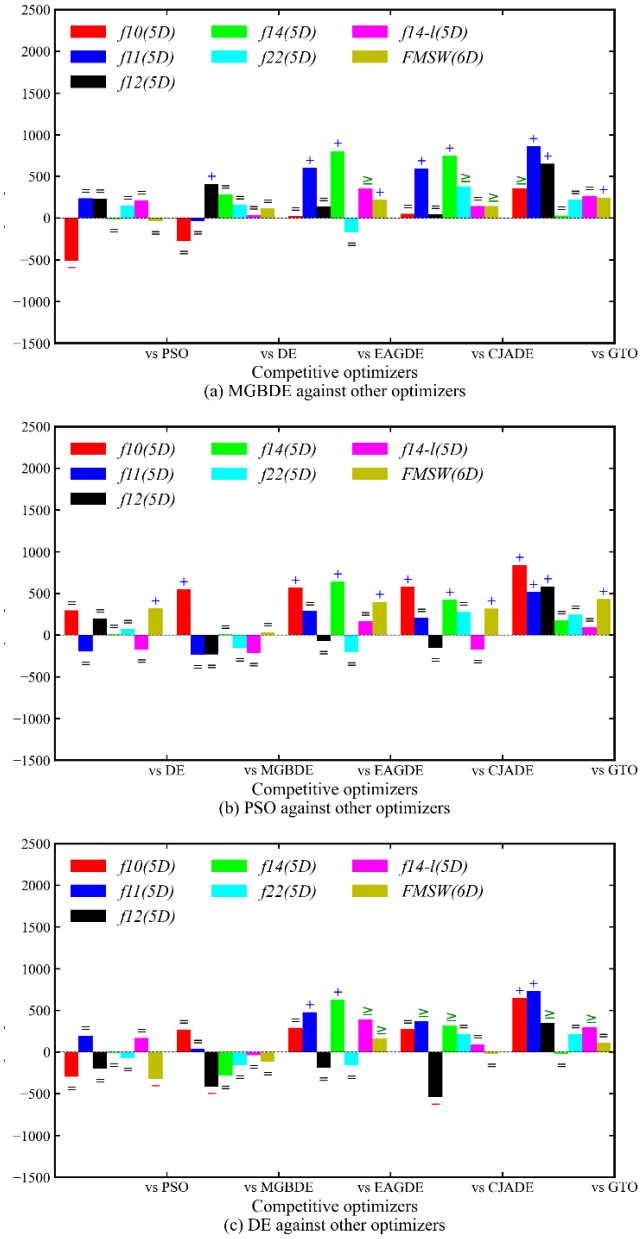


FIGURE 6. Wilcoxon signed-rank test for MGBDE, PSO and DE optimizer with other optimizers, respectively.

difference between the EI and PEI criterion. Additionally, although the PSO optimizer focuses more on rapid convergence and cannot always guarantee high-quality solutions on the PEI function, it still provides valuable algorithmic diversity to the proposed system. The diversity increases the probability of escaping local optima, leading to PSO's performance being second only to MGBDE. Furthermore, although EAGDE and CJADE demonstrate strong global search capabilities and often find high-quality solutions on the PEI function, they did not significantly improve the convergence performance of the proposed system. A potential reason is that finding high-quality solutions on the PEI function does not ensure an absolute improvement, as the PEI function is merely an approximation of the EI function, rather than the true EI function. Since the EI function is also based on the current surrogate model, high-quality solutions found on the PEI function may not lead to an improvement in the optimization objective, making it difficult to escape local optima. This limitation ultimately results in a reduction in convergence performance. Similarly, the GTO optimizer, which also possesses strong global search capabilities, demonstrates poor convergence performance in the proposed system due to reasons similar to those affecting EAGDE and CJADE.

TABLE 3. Mean rankings of optimizers for tested functions.

Ranking	Optimizer	Mean ranking	p value
1	MGBDE	1.8571	0.0062
2	PSO	2.4286	
3	DE	3.2857	
4	EAGDE	4.0000	
5	CJADE	4.0000	
6	GTO	5.4286	

B. COMPARATIVE INVESTIGATIONS ON BENCHMARK FUNCTIONS

1) EXPERIMENTAL SETTING

To evaluate the convergence performance of the proposed system in comparison with the traditional OK-based parallel EGO system, the MGBDE optimizer, identified as the most suitable for the proposed system, is employed to optimize the PEI function. Additionally, a gradient-based optimization algorithm, GS-SQP, is employed to tune the hyper-parameters of the GEK model. Table 4 provides descriptions of five benchmark functions and one engineering optimization problem, selected respectively from CEC2013 [51] and CLEC2011 [52], for performance evaluation. To maintain a reasonable computational cost, the feasible regions of all test functions were scaled to smaller ranges compared to the original intervals of $[-100, 100]^5$ and $[-6.4, 6.35]^6$. To reduce the potential impact of initial sample selection and optimizer randomness on the results, and to ensure experimental fairness, each system was evaluated 50 times for each test function. The 50 groups of initial samples, each consisting of 10D samples, are obtained using Latin hypercube designs (LHD) [53]. For each test case, a total of 200 updating points were added to the GEK model through the parallel EGO algorithm, with four points added in each cycle [46].

An object-oriented Python toolbox named Vehicular Aerodynamic Engineering Optimization in Python (Vaeopy) [54] was utilized to construct the corresponding kriging model and implement the parallel EGO algorithm. The parameter settings for constructing the OK and GEK models are specified as follows:

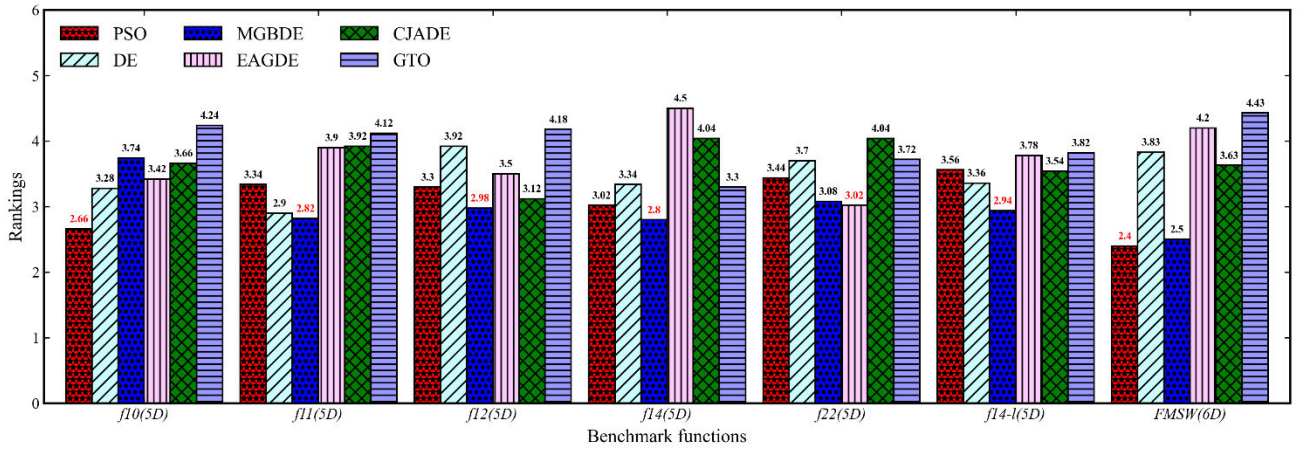


FIGURE 7. Friedman test between each optimizer for all tested functions.

- Regression function: *regpoly0*.
- Correlation function: *corrgauss*.
- Feasible region of hyper-parameters: $[10^{-5}, 10^{5}]$.

2) RESULTS AND DISCUSSION OF COMPARATIVE INVESTIGATIONS

A nonparametric statistical test [55] is employed to analyze significant differences among the comparative results. The Wilcoxon signed-rank test examines significant differences in convergence results between the two systems for each test function, with the significance level set at 0.05. A p-value below 0.05 indicates a significant performance difference between the two systems; otherwise, the difference is not significant.

To clearly illustrate the performance advantages of the proposed system, the optimization results obtained using the two comparative systems across all benchmark functions are compared. Figure 8 presents the Wilcoxon signed-rank test results, comparing the convergence levels of the proposed system with the traditional system across all selected test functions. The symbols "+" and "−" indicate that the first system outperforms or underperforms the second system, respectively. R^+ and R^- represent the sum of the rankings in the Wilcoxon signed-rank test, indicating that the first group of results outperforms or underperforms the second group, respectively. The y-axis in Figure 8 represents the difference between the absolute values of R^+ and R^-, with a larger gap between R^+ and R^- indicating a greater performance difference between the two compared systems. From the statistical results, it is clear that the convergence level of the GEK-based parallel EGO system is significantly better than that of the OK-based parallel EGO system on almost all tested benchmark functions, except for f10. The results show that by effectively utilizing additional gradient information to construct a high-reliability surrogate model, the updating point can acquire more effective information from the high-precision surrogate model, ensuring the search for higher-quality solutions through the PEI function.

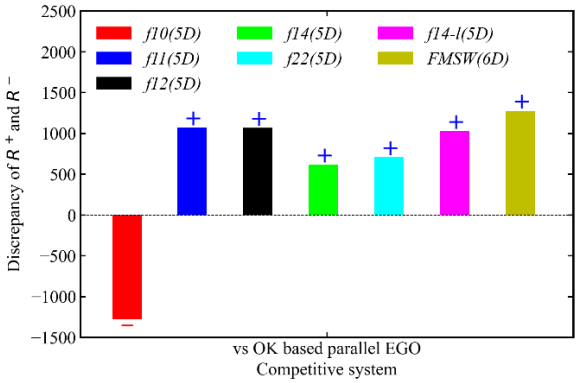


FIGURE 8. Wilcoxon signed-rank test between OK and GEK based parallel EGO system.

To observe the convergence trend of the entire optimization process more intuitively and better highlight the difference in convergence speed between the two compared systems, the average convergence curves of different systems on the test functions are presented in Figure 9. The GEK-based parallel EGO system converged significantly faster than the OK-based parallel EGO system on nearly all tested functions, except for f10. Notably, on the FMSW optimization problem, the OK-based parallel EGO system achieved minimal improvement throughout the optimization process, while the GEK-based parallel EGO system effectively addressed this type of optimization problem.

C. COMPARATIVE INVESTIGATION ON REAL-WORLD OPTIMIZATION PROBLEM OF AIRFOIL AERODYNAMICS

The proposed GEK-based parallel EGO system is employed to address a real-world airfoil aerodynamic optimization problem, with its optimization performance compared to that of the OK-based parallel EGO system. To accomplish this, key details of the optimization problem are outlined,

TABLE 4. Details of benchmark test functions.

Benchmark function	Property	Lower bound	Upper bound	optimum
Rotated Griewank's Function (*f*10)	Multi-modal; Rotated; Non-separable	[-50, 0, -60, 30, -50]	[0, 50, -10, 80, 0]	-500
Rastrigin's Function (*f*11)	Multi-modal; Separable; Asymmetrical	[-50, 0, -60, 30, -50]	[0, 50, -10, 80, 0]	-400
Rotated Rastrigin's Function (*f*12)	Multi-modal; Non-separable; Asymmetrical	[-50, 0, -60, 30, -50]	[0, 50, -10, 80, 0]	-300
Schwefel's Function (*f*14; *f*14-l)	Multi-modal; Rotated; Non-separable; Asymmetrical	[-50, 0, -60, 30, -50], [-80, -30, -90, 0, -80]	[0, 50, -10, 80, 0], [20, 70, 10, 100, 20]	-100
Composition Function (*f*22)	Multi-modal; Separable; Asymmetrical	[-50, 0, -60, 30, -50]	[0, 50, -10, 80, 0]	800
Frequency Modulated Sound Waves (FMSW)	Multi-Modal; Strong hierarchy	[0, 4, -2.5, 3.8, 1, 3.9]	[2, 6, -0.5, 5.8, 3, 5.9]	0

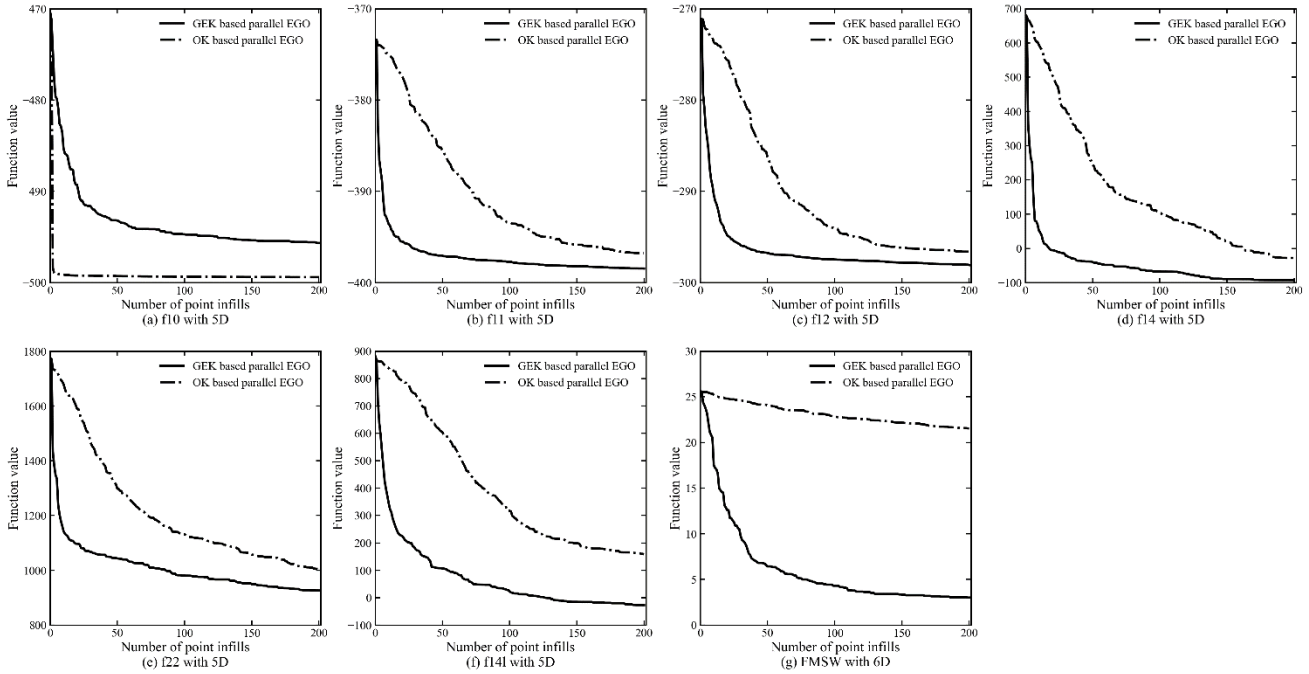


FIGURE 9. Average convergence curves of comparative systems on benchmark tested function.

including the airfoil model, optimization objective, design variables, meshing strategy, and computational fluid dynamics (CFD) solver. The rationality of the numerical simulation was first validated, followed by a detailed discussion of the optimization performance of the proposed system in addressing the airfoil shape optimization problem.

1) DETERMINATION OF AIRFOIL

The NACA0012, a classic symmetrical airfoil widely utilized in low Reynolds number flow and control studies [56], [57], serves as the baseline model for shape optimization. In this study, two-dimensional numerical simulations are conducted for airfoil shape optimization. The airfoil model has a relative thickness of 12%, with the maximum thickness positioned at 40% of the chord length from the leading edge. The chord length is set to 1 meter, as illustrated in Figure 10.

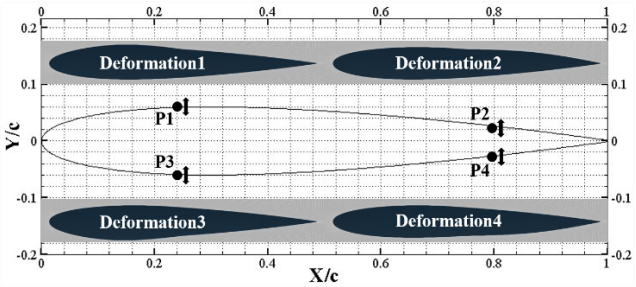


FIGURE 10. Baseline model of NACA0012 airfoil and definition of design variables.

2) DETERMINATION OF OBJECTIVE AND DESIGN VARIABLES

This study selects the maximization of the lift-to-drag ratio (Cl/Cd) relative to the baseline airfoil model as the

TABLE 5. Details of design variables.

Deformation position	Variable	Lower bound (mm)	Upper bound (mm)
23.62% (upper edge)	P1	-20	+20
76.38% (lower edge)	P2	-20	+20
23.62% (upper edge)	P3	-20	+20
76.38% (lower edge)	P4	-20	+20

optimization objective. To enhance the aerodynamic performance of the airfoil while minimizing computational costs, the design variables are carefully selected to modify the shape of the airfoil model. Yu et al. [58] reported that the aerodynamic performance of an airfoil is predominantly affected by the pressure differences between its leading and trailing edges, as well as between the upper and lower surfaces. Therefore, four control points were defined as design variables, located at 23.62%c and 76.38%c along both the upper and lower surfaces of the airfoil, measured from the leading edge.

To control the airfoil shape automatically, the Direct Morphing (DFM) module in ANSA v23 is utilized to perform shape deformation based on the original airfoil configuration. Figure 10 illustrates the definition of each design variable and the corresponding deformed models. The deformation process is managed by adjusting the positions of control points, represented by black solid circles, along the y-direction. Table 5 specifies the feasible range for each design variable. To maintain the integrity of the airfoil shape after deformation, appropriate upper and lower bounds are established for the design variables.

3) NUMERICAL SIMULATION SETTING AND VERIFICATION

To obtain the Cl/Cd of the airfoil models before and after deformation, CFD solver Fluent in ANSYS 2021 R1, a commercial CFD software, running on the computer along with the $K-\omega-SST$ turbulence model was used to perform the numerical simulation. The computer is equipped with an i9-12900K processor, which operates at a base frequency of 3.20 GHz and has 16 cores. The details of the computational domain and mesh generation are as follows: the distance between the far-field boundary of the computational domain and the airfoil surface is set to 50c [59]. A C-H type structured grid is adopted as the mesh strategy. To capture the details of the flow field around the airfoil, the mesh has a growth rate of 1.1, and the first layer height is set to 0.015 mm to ensure that the y+ value is controlled to be less than 1, which is necessary for the turbulence model used [60]. The total number of mesh elements is 106,752.

Reasonable boundary conditions and solver settings are crucial for accurately solving the external flow field around the airfoil. In this study, the inlet velocity is set to 42.0245m/s

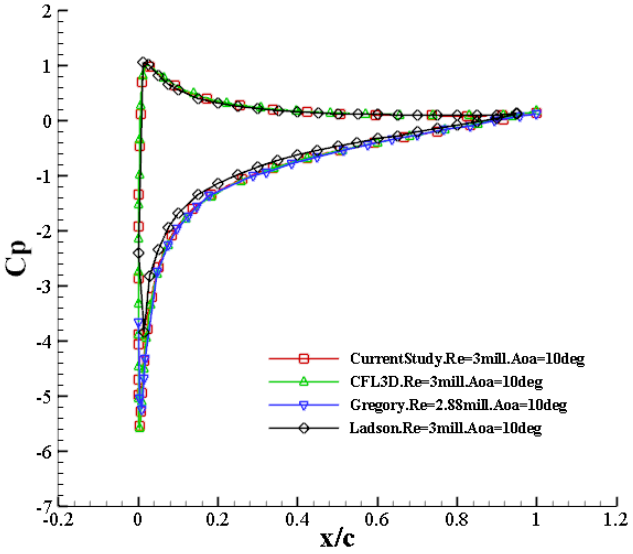


FIGURE 11. Validation of the Cp value of airfoil surface with previous studies [61].

TABLE 6. Validation of aerodynamic characteristics of Cl and Cd of airfoil with previous studies [61].

Data source	Cl	Cd
In this study	1.04994	0.01590
CFL3D	1.07781	0.01236
Gregory	0.98488	--
Ladson	1.07070	0.01201

($\mathrm{Re} = 3.0\times10^6$), and the incoming flow angle of attack is 10°. The outlet pressure is fixed at zero. The airfoil surface boundary is set to a no-slip condition. The simulation is conducted using the semi-implicit method of pressure-linked equations (SIMPLE) algorithm with GAME for pressure and ''smoothSolver'' for the transport parameters. The number of iterations is set to 500 for each simulation, and the convergence tolerance for each solution variable is set to 1×10^{-6}.

To verify the reliability of the numerical simulation method, the results obtained from the original model used in this study are compared with those from previous studies [61] (NASA Langley Turbulence Modeling Resource website). The comparative results are shown in Figure 11. It is evident that the variation trend of the pressure coefficient (Cp) on the airfoil surface in this study aligns closely with the results from previous studies. Additionally, Table 6 presents the Cl and Cd values obtained in this and previous studies. The computational error is within acceptable limits. In conclusion, the numerical simulation method employed in this study is reliable and suitable for subsequent optimization studies.

4) OPTIMIZATION SETTING

To efficiently apply the proposed optimization system for airfoil shape optimization, a series of key processes must be performed sequentially. These processes include sampling, evaluation, construction of the GEK surrogate model, and the point infill criterion to drive global convergence during the optimization process. Five initial sample points were obtained using the LHD sampling method. Numerical simulations were conducted to obtain the Cl and Cd values for each sample point. The OK and GEK-based parallel EGO algorithm systems were repeatedly run until a stopping criterion was met, respectively. It is important to note that real-world optimization problems are often constrained by development time and high costs. Thus, achieving the maximum possible improvement within a limited time, rather than focusing solely on obtaining a global optimum, better aligns with the practical needs of optimization in real-world scenarios. In consideration of this, a total of 20 point infills (4 updating points per cycle) were selected to evaluate the performance of the proposed system. Another important aspect to note is that the Cl/Cd value of the original airfoil model was 66.03396, while the maximum Cl/Cd value in the five initial samples was 70.07862. Therefore, the actual improvement in the Cl/Cd value during the optimization process is measured based on the maximum objective value from the initial samples, meaning that a higher Cl/Cd value than 70.07862 must be achieved. Additionally, the required gradient information for constructing the GEK is obtained using the finite difference method. The differential step length is set to 1×10^{-3}m, which yielded the best results in the tests. The specific settings for constructing the OK and GEK models are the same as those described in Section V-B.

5) RESULTS AND DISCUSSION OF COMPARATIVE INVESTIGATIONS

To comprehensively evaluate the performance advantages of the proposed optimization system, the improvement in the Cl/Cd ratio with different systems as the number of point infills increases is presented in Figure 12. The proposed system converges more efficiently than the traditional system. Notably, the former system identified a high-quality solution at the first updating point and achieved continuous improvement in the Cl/Cd ratio, increasing it from 70.07862 to 72.04189 throughout nearly the entire optimization process. In contrast, the latter system did not identify an improved solution during the first point infill cycle and showed only limited improvement over the first 12 updating points. Table 7 presents the details of the optimal solutions obtained by two compared systems. The final improvement in the Cl/Cd ratio achieved by the proposed system and the traditional system was 2.80% and 2.39%, respectively, compared to the baseline airfoil model. To better understand the mechanism behind the improvement in the Cl/Cd ratio following airfoil shape optimization, a comparative analysis of the pressure distribution on the airfoil surface before and after optimization was conducted. Figure 13 illustrates the surface pressure distribution for both the baseline and optimized airfoils using different optimization systems. The optimal solutions obtained by both systems resulted in significant positive changes in the control variables P1, P3, and P4 along the y-direction. As a result, the pressure around these control points also underwent substantial changes compared to the baseline airfoil. The primary reason for the improvement in the Cl/Cd ratio is the enhanced pressure difference between the upper and lower surfaces of the airfoil, which significantly increases the lift generated by the airfoil. To more intuitively observe the flow field changes resulting from airfoil shape modifications during the optimization process, Figure 14 presents the corresponding pressure contours for the baseline airfoil model and the optimized airfoil models at the 1st, 7th, 13th, and 19th solutions using different optimization systems. The pressure contours clearly show that significant changes in the control points correspond to notable variations in the pressure distribution.

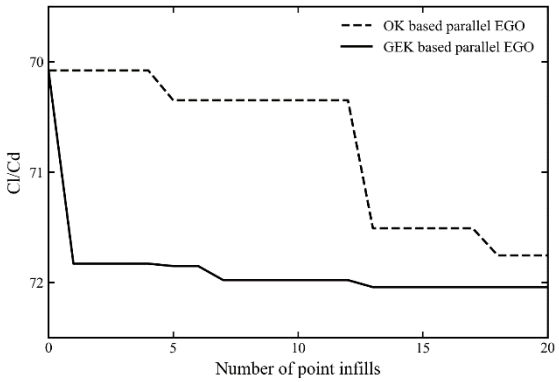


FIGURE 12. Convergence curves of Cl/Cd ratio with GEK and OK based parallel EGO algorithm system, respectively.

D. COMPARATIVE INVESTIGATION ON REAL-WORLD OPTIMIZATION PROBLEM OF VEHICLE AERODYNAMICS

To demonstrate the performance advantages of the proposed algorithm system in more complex engineering optimization problems, this section presents a comparative investigation of another real-world optimization problem in vehicle aerodynamics.

1) ACQUISITION OF HIGH-FIDELITY DATASETS

To efficiently and rapidly validate the performance advantages of the proposed improved algorithm system in automotive aerodynamic optimization, a high-precision neural network model [62], trained on a large high-fidelity dataset, was employed to evaluate the objective. The dataset for the high-fidelity model was constructed using aerodynamic parameters, such as drag, lift, and changed drag, of a full-scale SUV vehicle shape. The baseline vehicle shape and the 10 deformed vehicle shapes, obtained through parametric deformation using the commercial pre-processing software "DEP MeshWorks," are shown in Figure 15. A total of

TABLE 7. Optimal solutions obtained by gek and ok based parallel algorithm system.

System	P1 (mm)	P2 (mm)	P3 (mm)	P4 (mm)	Cl	Cd	Cl/Cd
GEK based parallel EGO	19.37	2.90	20.00	20.00	1.18653	0.01647	72.04189
OK based parallel EGO	16.21	-2.97	20.00	20.00	1.16959	0.01630	71.75399

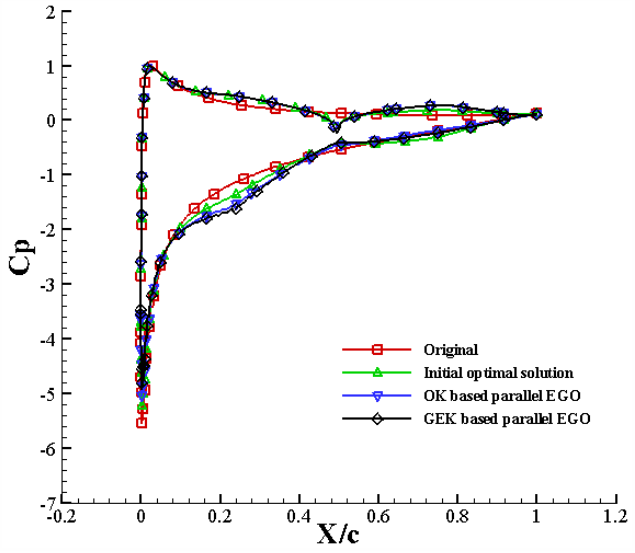


FIGURE 13. Surface pressure coefficient distribution of the baseline airfoil model and the optimized airfoil model using GEK and OK based parallel algorithm system, respectively.

29 design variables were selected to control the vehicle shape. A total of 232 and 123 sample points were chosen from the design space using the OLHD method and the optimization algorithm, respectively. These objective values were efficiently and accurately evaluated using the 'Fugaku' supercomputer [63] and the 'CUBE' numerical simulation method [64]. In this case, the inlet velocity was set to 27.78 m/s toward the vehicle model, and the Reynolds number was 5.83×10^6Pa · s based on the original vehicle length, inlet velocity, and fluid viscosity. The numerical simulation was conducted using the ''Fractional-step method,'' ''Crank-Nicolson method,'' and ''Quick scheme.'' For grid resolution, the minimum grid size was 7 mm, and the total number of grids was 201 million. The total numerical simulation time was 2 s, with the time step set to 2.5×10^{-5}s.

2) CONSTRUCTION OF HIGH-PRECISION NEURAL NETWORK MODEL

A fully connected neural network model was trained using 355 high-fidelity original datasets, with the 'ReLU' activation function applied. Different training parameters were compared to determine the optimal configuration. Both the input layer and hidden layer were set to 30 nodes, which provided the best performance. The output layer consisted of 3 nodes, each representing a predicted objective value for the aerodynamic performance of the vehicle.

The predictive accuracy and reliability of the neural network model were validated using 10 vehicle models, as mentioned in the previous section. In this study, only the Cd value is considered as the optimization objective for validating the performance of the proposed algorithm system. Therefore, Figure 16 shows the validation of the predicted Cd values from the neural network model and the numerical simulation experiment. It can be observed that the predicted Cd values align well with the numerical simulation results. Both experimental and predicted Cd values were normalized. The average prediction error is 0.52%, which is within the acceptable range for practical engineering applications.

3) OPTIMIZATION PROCESS AND RESULTS ANALYSIS

In the current study, only the Cd value is considered as the optimization objective for validating the performance of the proposed algorithm system. Therefore, when evaluating the objective values of the sample points using the constructed neural network model, the focus is solely on the Cd value. The LHD method was used to select 10 initial sample points from the design space, which were then used to construct OK and GEK surrogate models after obtaining the Cd value of each sample point. To obtain the gradient information required for constructing the GEK surrogate model, two methods were employed. The first method is the finite difference method (with a step size of 0.001). The second method leverages the training characteristics of the neural network model, enabling the efficient acquisition of gradient information for each dimension of each sample point through automatic differentiation techniques. Figure 17 presents the gradient information obtained using both methods. As shown, the gradient information derived from the two approaches is generally consistent. During the optimization process, each of the compared algorithm systems filled 60 new sample points, with 4 new sample points added in each iteration, following the parallel point infill criterion.

Figure 18 presents the normalized convergence curves of the parallel EGO optimization algorithm systems based on the OK and GEK surrogate models, respectively. From the figure, it can be observed that the parallel EGO algorithm system based on the OK surrogate model fails to find an improved solution during the early optimization stage due to

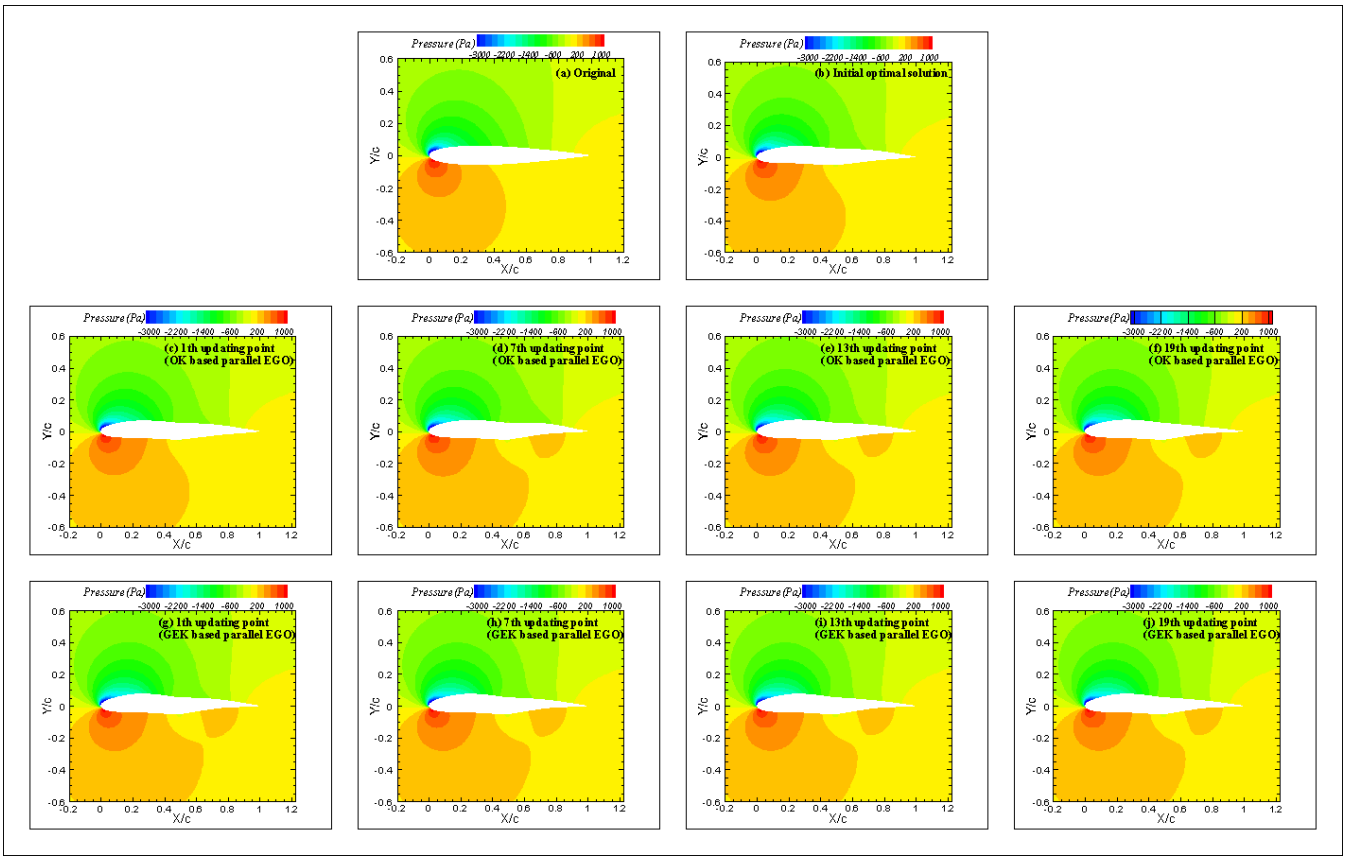


FIGURE 14. Pressure contour of the baseline airfoil model and the optimized airfoil model using GEK and OK based parallel algorithm system, respectively.

the limited effective information provided. It only identifies the first improved solution after the 32nd sample is added, resulting in a relatively slow convergence speed. In contrast, the system based on the GEK surrogate model is able to identify an improved solution even in the early optimization stage, leveraging the limited information available. Moreover, as more sample points are added, the GEK-based system rapidly and consistently converges towards the optimal objective. Notably, within the current limited number of sample points, the GEK-based system ultimately converges to higher-quality solutions, while the OK-based system converges to a lower-quality solution. Specifically, the final maximum drag reduction rate achieved is 12.97%, relative to the initial optimum.

E. QUANTITATIVE ANALYSIS OF THE PROPOSED METHOD AND THE TRADITIONAL METHOD ON REAL-WORLD OPTIMIZATION PROBLEM OF AERODYNAMICS

To validate the performance advantages of the proposed improved parallel EGO optimization system in practical engineering optimization problems, experimental verification was conducted in Sections C and D. From the convergence curves in Figures 12 and 18, it was observed that the proposed method achieves faster convergence than the traditional parallel EGO optimization system. Additionally, under the same number of sample point infills, the proposed method identifies higher-quality approximate optimal solutions.

To further analyze its performance advantages, a quantitative comparison was conducted in terms of samples cost, optimization cycles, computational resource cost, and total wall-clock time cost required to reach a predefined optimization target. Firstly, it is essential to understand the main components of wall-clock time cost. In our case, the wall-clock time cost for a parallel infill process consists of three parts: mesh preparation, waiting gap, and simulation, as shown in Figure 19. It is worth noting that updating the surrogate model and searching for new sample points based on the infill criterion require significantly less time than other steps, making their wall-clock time cost negligible. Specifically, the wall-clock time cost per parallel infill process for the airfoil and vehicle shape optimization problems is 0.33 h and 14.5 h, respectively. Table 8 shows a quantitative comparison of optimization costs between the proposed improved optimization system and the traditional parallel EGO optimization system. In the airfoil shape optimization problem, the traditional parallel EGO optimization system requires 18 samples infills, corresponding to 5 parallel optimization cycles, to reach the optimization target. In contrast, the GEK-based parallel EGO optimization system achieves

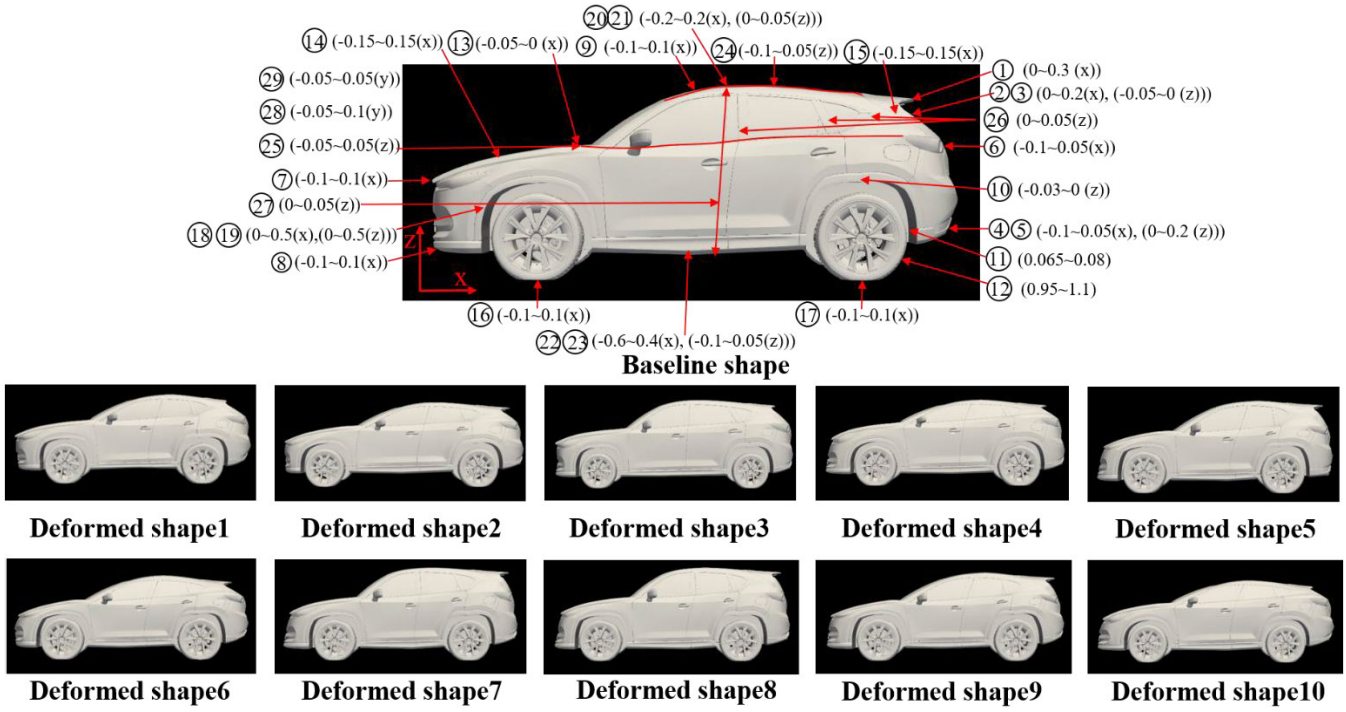


FIGURE 15. Baseline vehicle shape, and deformed 10 vehicle shapes based on definition of design variables.

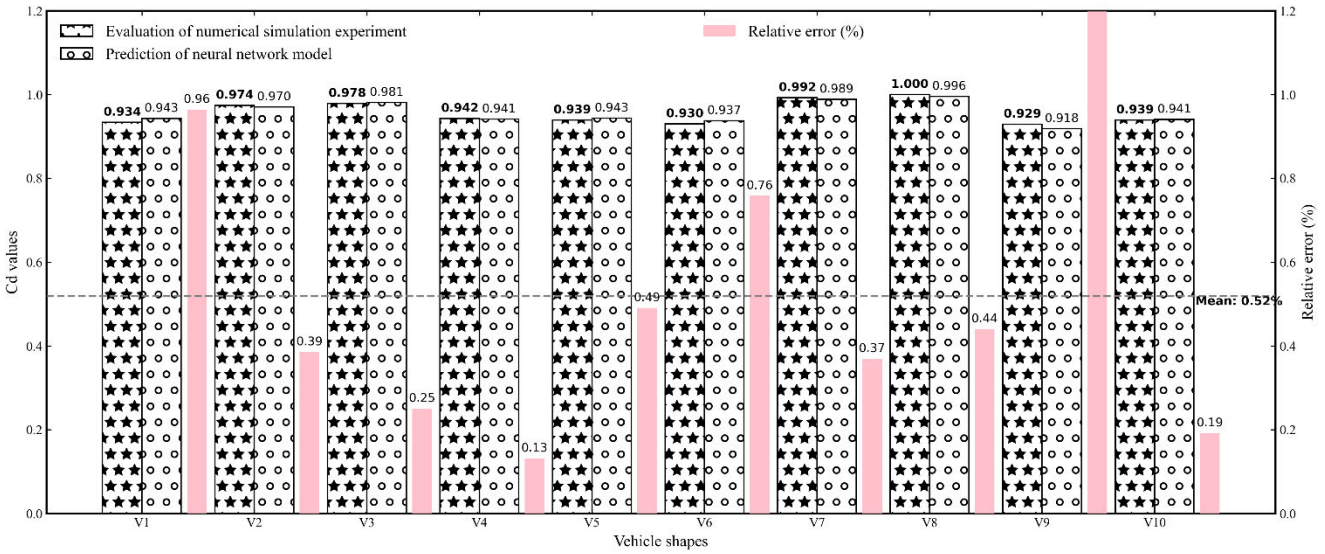


FIGURE 16. Validation of the predicted Cd values of the neural network model and numerical simulation experiment, respectively.

the same target with only one samples infill and a single optimization cycle. Furthermore, the GEK-based system has a lower wall-clock time cost. For the vehicle shape optimization problem, the traditional parallel EGO optimization system requires 56 samples infills, equivalent to 14 optimization cycles, whereas the GEK-based system reaches the target with only 2 samples infills and a single optimization cycle. Additionally, the GEK-based system significantly reduces the wall-clock time cost compared to the traditional parallel EGO optimization system.

F. POTENTIAL LIMITATIONS OF THE PROPOSED OPTIMIZATION SYSTEM

Performance study of the proposed system on real-world engineering problems demonstrate its advantages over the traditional parallel EGO system in terms of convergence

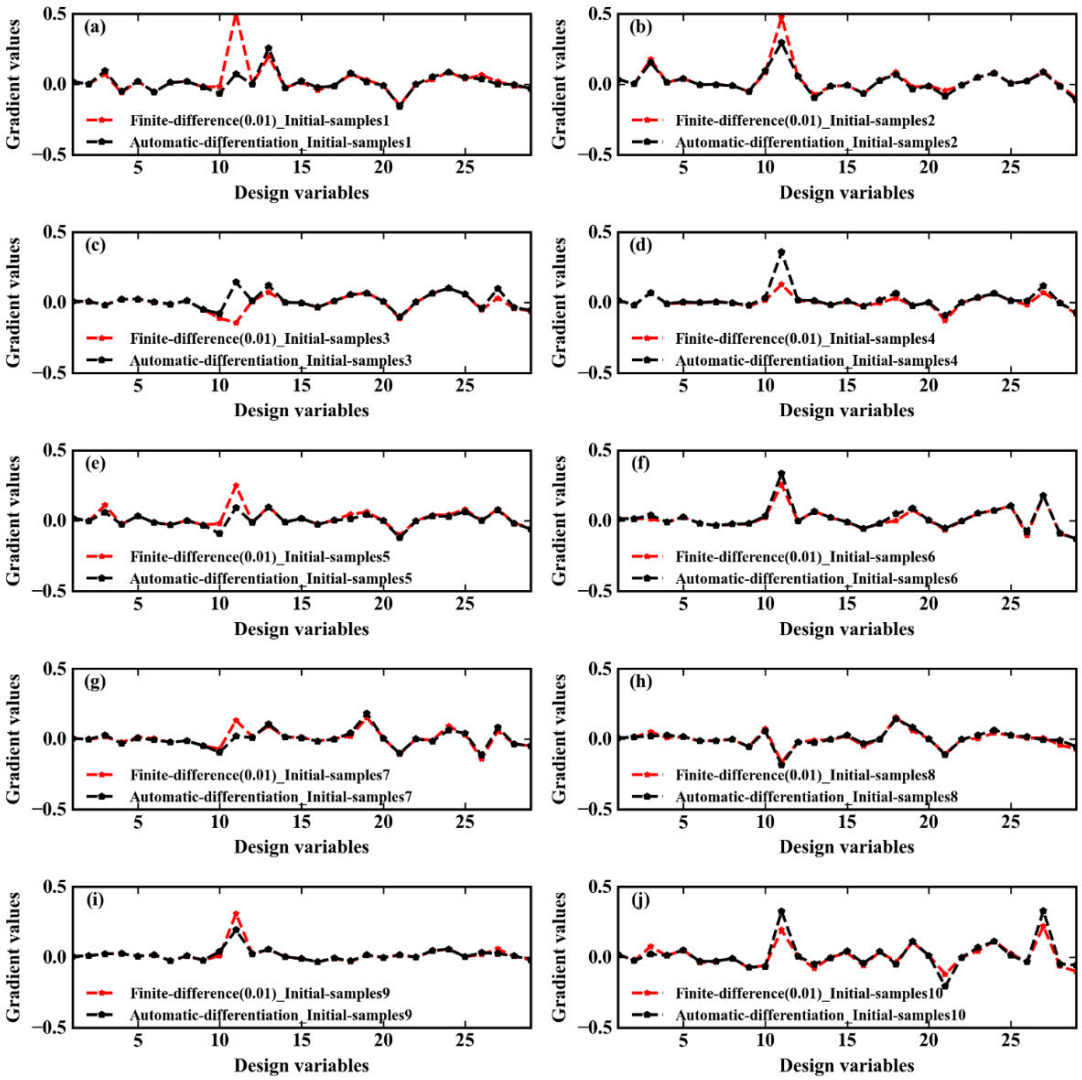


FIGURE 17. Validation of the obtained gradient information of finite difference and automatic differentiation technique, respectively.

TABLE 8. Quantitative comparison of performance of the GEK and OK based parallel algorithm system in optimization process.

Optimized target	Improved aerodynamic performance（Case C: 2.4%）				Improved aerodynamic performance（Case D: 7.0%）			
Performance Indicators	Cost of samples	Execute optimization cycles	Cost of evaluation objective	Cost of Wall-clock time (*h*)	Cost of samples	Execute optimization cycles	Cost of evaluation objective	Cost of Wall-clock time (*h*)
OK based parallel EGO	18	5	18	1.65	56	14	56	203
GEK based parallel EGO	1	1	1	0.33	2	1	2	14.5

speed, solution quality, and wall-clock time cost. However, certain limitations remain.

These limitations can be categorized into two types. The first arises when the optimization problem exhibits strong nonlinearity, leading to noise in numerical simulation results. This noise may cause overfitting in the GEK model, restricting its generalization ability. Fortunately, this issue can be mitigated by adopting the concept of regression kriging [31], which filters out noise from numerical simulations, resulting in a smoother and more generalizable surrogate model. The

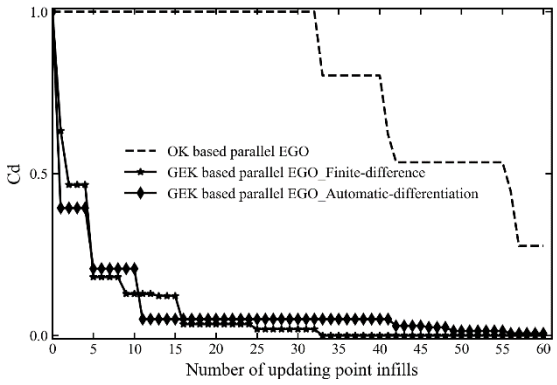


FIGURE 18. Convergence curves of Cd values with GEK and OK based parallel EGO algorithm system, respectively.

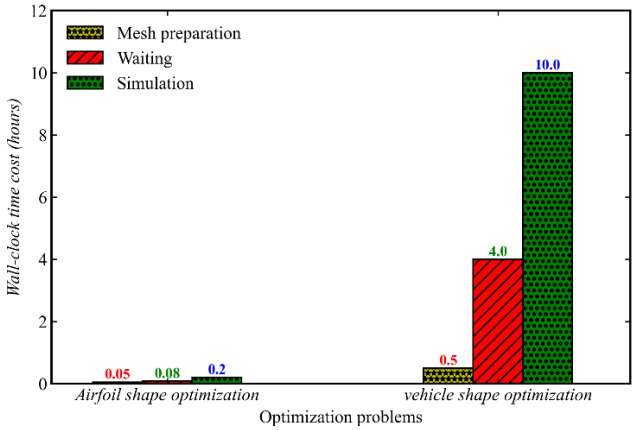


FIGURE 19. Wall-clock time cost components of each optimization cycle.

second limitation stems from the additional computational cost required to obtain gradient information for the initial GEK model. When the number of design variables is high, obtaining gradients via traditional finite difference method can become computationally prohibitive. Overcoming challenge requires improving gradient computation efficiency. Fortunately, the adjoint method in simulations offers an effective solution [65], enabling gradient computation at a cost of approximately twice that of a standard simulation, regardless of the problem's dimensionality. This significantly enhances the proposed method's applicability to high-dimensional optimization problems. In summary, addressing these limitations will further increase the practical value of the proposed system for engineering applications.

VI. CONCLUSION

In this study, to fully exploit the performance advantages of the parallel EGO algorithm, an improved algorithm system called the GEK-based parallel EGO algorithm system was proposed. This system is derived from the OK-based parallel EGO algorithm system by introducing a powerful surrogate model called GEK. It effectively utilizes gradient information to enhance the construction of a high-fidelity surrogate model, improving the reliability of the EGO algorithm in identifying potential improvement points based on the parallel infill criterion. Additionally, optimization performance tests were conducted on benchmark functions to ensure the robustness of the proposed system and identify the most suitable optimizer. Based on this, performance comparisons and analyses were conducted between the proposed method and the traditional parallel EGO optimization system on benchmark functions and two aerodynamic shape optimization applications. From the comparison of results, the following conclusions were drawn:

1) The hybrid MGBDE optimizer, with its stronger balance between exploration and exploitation, achieved the best performance on the proposed algorithm system. Therefore, for real-world engineering optimization problems, it is recommended that the MGBDE optimizer be used as the core engine to search for potential improvement points.
2) The proposed improved system demonstrates significant improvement in convergence performance compared to the original system on nearly all tested benchmark functions and two real-world engineering optimization problems.
3) Quantitative analysis of airfoil and automotive shape optimization cases demonstrates that the proposed algorithm system significantly reduces wall-clock time compared to traditional methods, highlighting its potential for real-world engineering applications.

Notably, while the proposed system significantly improves optimization performance over the traditional parallel EGO system, certain limitations must be addressed for practical engineering applications, as discussed in Section V-F. Additionally, future work will explore integrating deep learning-based surrogate models and hardware acceleration strategies to enhance computational efficiency, ultimately applying these advancements to real-world engineering optimization problems.

ABBREVIATIONS

EGO	Efficient Global Optimization, a global optimization algorithm.
PEI	Pseudo Expected Improvement, a point-infilling criterion.
GEK	Gradient-enhanced Kriging, a surrogate model.
OK	Ordinary Kriging, a surrogate model.
SBO	Surrogate-based Optimization, an optimization technology.
EI	Expected Improvement, a point-infilling criterion.
q-EI	q-point Expected Improvement, a point-infilling criterion.
KB	Kriging Believer, a point-infilling criterion.
CL	Constant Liar, a point-infilling criterion.

IF	Influence Functions, a series of correlation functions.
GS-SQP	Golden Search Sequential Quadratic Programming, a gradient based optimization algorithm.
DE	Differential Evolution, an evolutionary optimization algorithm.
PSO	Particle Swarm Optimization, a population based optimization algorithm.
EAGDE	Enhanced Adaptive Guided Differential Evolution, an evolutionary optimization algorithm.
CJADE	Chaotic local search-based JADE, an evolutionary optimization algorithm.
MGBDE	Modified Gaussian Bare-bones Differential Evolution, an evolutionary optimization algorithm.
GTO	Artificial Gorilla Troop Optimizer, a nature-inspired optimization algorithm.
LHD	Latin Hypercube Designs, a sampling method.
Vaeopy	Vehicular aerodynamic engineering optimization in Python, an optimization toolbox in Python.
CFD	Computational Fluid Dynamics, a numerical simulation technology.
Cl	Lift Coefficient, a dimensionless parameter that describes the forces acting on an object in fluid dynamics.
Cd	Drag Coefficient, a dimensionless parameter that describes the forces acting on an object in fluid dynamics.
DFM	Design for Manufacturing, a morphing technology in ANSA software.
SIMPLE	Semi-implicit Method of the Pressure-linked Equations, a numerical solution algorithm.
Cp	Pressure Coefficient, a dimensionless parameter that describes the pressure distribution in fluid dynamics.
OLHD	Optimal Latin Hypercube Designs, a sampling method.
CUBE	Complex Unified Building cubE method, a scalable framework for large-scale industrial simulations.

REFERENCES

[1] M. Bunge, "A general black box theory," *Philosophy Sci.*, vol. 30, no. 4, pp. 346–358, Oct. 1963.

[2] W. H. Hucho, L. J. Janssen, and H. J. Emmelmann, "The optimization of body details—A method for reducing the areodynamic drag of road vehicles," *SAE Trans.*, vol. 85, pp. 760163–760429, Feb. 1976.

[3] P. N. Krishnani and D. Zhou, "CFD analysis of drag reduction for a generic SUV," in *Proc. ASME Int. Mech. Eng. Congr. Expo.*, Lake Buena Vista, FL, USA, Nov. 2009, pp. 589–598.

[4] Y. Liu, J. Chen, J. Cheng, and H. Xiang, "Aerodynamic optimization of transonic rotor using radial basis function based deformation and data-driven differential evolution optimizer," *Aerospace*, vol. 9, no. 9, p. 508, Sep. 2022.

[5] T. Kojima, N. A. V. Doan, and H. Amano, "GenMap: A genetic algorithmic approach for optimizing spatial mapping of coarse-grained reconfigurable architectures," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 11, pp. 2383–2396, Nov. 2020.

[6] Z. He, T. Liu, and H. Liu, "Improved particle swarm optimization algorithms for aerodynamic shape optimization of high-speed train," *Adv. Eng. Softw.*, vol. 173, Nov. 2022, Art. no. 103242.

[7] Z. A. Ali, Z. Han, and R. J. Masood, "Collective motion and self-organization of a swarm of UAVs: A cluster-based architecture," *Sensors*, vol. 21, no. 11, p. 3820, May 2021.

[8] A. K. Mohamed and A. W. Mohamed, "Real-parameter unconstrained optimization based on enhanced AGDE algorithm," in *Machine Learning Paradigms: Theory and Application* (Studies in Computational Intelligence). Cham, Switzerland: Springer, 2019, pp. 431–450.

[9] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou, "Chaotic local search-based differential evolution algorithms for optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 6, pp. 3954–3967, Jun. 2021.

[10] H. Wang, S. Rahnamayan, H. Sun, and M. G. H. Omran, "Gaussian bare-bones differential evolution," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 634–647, Apr. 2013.

[11] B. Abdollahzadeh, F. Soleimanian Gharehchopogh, and S. Mirjalili, "Artificial gorilla troops optimizer: A new nature-inspired Metaheuristic algorithm for global optimization problems," *Int. J. Intell. Syst.*, vol. 36, no. 10, pp. 5887–5958, Jul. 2021.

[12] M. Shafiq, Z. A. Ali, A. Israr, E. H. Alkhammash, M. Hadjouni, and J. J. Jussila, "Convergence analysis of path planning of multi-UAVs using max-min ant colony optimization approach," *Sensors*, vol. 22, no. 14, p. 5395, Jul. 2022.

[13] A. Slowik and H. Kwasnicka, "Evolutionary algorithm and their application to engineering problems," *Neural. Comput. Applic.*, vol. 32, pp. 12363–12379, Aug. 2020.

[14] M. Urquhart, E. Ljungskog, and S. Sebben, "Surrogate-based optimisation using adaptively scaled radial basis functions," *Appl. Soft Comput.*, vol. 88, Mar. 2020, Art. no. 106050.

[15] T. W. Simpson, J. D. Poplinski, P. N. Koch, and J. K. Allen, "Metamodels for computer-based engineering design: Survey and recommendations," *Eng. Comput.*, vol. 17, no. 2, pp. 129–150, Jul. 2001.

[16] V. Raul and L. Leifsson, "Surrogate-based aerodynamic shape optimization for delaying airfoil dynamic stall using Kriging regression and infill criteria," *Aerosp. Sci. Technol.*, vol. 111, Apr. 2021, Art. no. 106555.

[17] B. Liu, H. Liang, Z.-H. Han, and G. Yang, "Surrogate-based aerodynamic shape optimization of a sliding shear variable sweep wing over a wide mach-number range with plasma constraint relaxation," *Structural Multidisciplinary Optim.*, vol. 66, no. 43, p. 43, Feb. 2023.

[18] G. Demir, R. M. Gorguluarslan, and S. Aradag, "Robust shape optimization under model uncertainty of an aircraft wing using proper orthogonal decomposition and inductive design exploration method," *Struct. Multidisc. Optim.*, vol. 66, no. 4, p. 93, Apr. 2023.

[19] A. Ariyarit, M. Sugiura, Y. Tanabe, and M. Kanazaki, "Hybrid surrogate-model-based multi-fidelity efficient global optimization applied to helicopter blade design," *Eng. Optim.*, vol. 50, no. 6, pp. 1016–1040, Sep. 2017.

[20] M. Urquhart, M. Varney, S. Sebben, and M. Passmore, "Aerodynamic drag improvements on a square-back vehicle at yaw using a tapered cavity and asymmetric flaps," *Int. J. Heat Fluid Flow*, vol. 86, Dec. 2020, Art. no. 108737.

[21] Z. He, X. Xiong, B. Yang, and H. Li, "Aerodynamic optimisation of a high-speed train head shape using an advanced hybrid surrogate-based nonlinear model representation method," *Optim. Eng.*, vol. 23, no. 1, pp. 59–84, Mar. 2022.

[22] S. Tinmitondé, X. He, and L. Yan, "Single-objective aerodynamic optimization of a streamlined bridge deck subjected to shape modification using a polynomial emulator and genetic algorithm," *Structural Multidisciplinary Optim.*, vol. 65, no. 12, Nov. 2022.

[23] X. Zhou, W. Guo, X. Shi, C. She, Z. Zheng, J. Zhou, and Y. Zhu, "Machine learning assisted multi-objective design optimization for battery thermal management system," *Appl. Thermal Eng.*, vol. 253, Sep. 2024, Art. no. 123826.

[24] M. G. M. Abdolrasol, S. M. S. Hussain, T. S. Ustun, M. R. Sarker, M. A. Hannan, R. Mohamed, J. A. Ali, S. Mekhilef, and A. Milad, "Artificial neural networks based optimization techniques: A review," *Electron.*, vol. 10, no. 21, p. 2689, Nov. 2021.

[25] W. Zhang, X. Gu, L. Tang, Y. Yin, D. Liu, and Y. Zhang, "Application of machine learning, deep learning and optimization algorithms in geoengineering and geoscience: Comprehensive review and future challenge," *Gondwana Res.*, vol. 109, pp. 1–17, Sep. 2022.

[26] P. Wu, W. Yuan, L. Ji, L. Zhou, Z. Zhou, W. Feng, and Y. Guo, "Missile aerodynamic shape optimization design using deep neural networks," *Aerosp. Sci. Technol.*, vol. 126, Jul. 2022, Art. no. 107640.

[27] J. Li, M. Zhang, J. R. R. A. Martins, and C. Shu, "Efficient aerodynamic shape optimization with deep-learning-based geometric filtering," *AIAA J.*, vol. 58, no. 10, pp. 4243–4259, Oct. 2020.

[28] H. Hu, G. Zhang, D. Li, Z. Zhang, T. Sun, and Z. Zong, "Shape optimization of airfoil in ground effect based on free-form deformation utilizing sensitivity analysis and surrogate model of artificial neural network," *Ocean Eng.*, vol. 257, Aug. 2022, Art. no. 111514.

[29] G. G. Wang and S. Shan, "Review of metamodeling techniques in support of engineering design optimization," *J. Mech. Design*, vol. 129, no. 4, pp. 370–380, Apr. 2007.

[30] A. Booker, "Design and analysis of computer experiments," *Statist. Sci.*, vol. 4, no. 4, pp. 409–423, Aug. 1989.

[31] A. I. Forrester, A. J. Keane, and N. W. Bressloff, "Design and analysis of 'noisy' computer experiments," *AIAA J.*, vol. 44, no. 10, pp. 2331–2339, Oct. 2006.

[32] J. Laurenceau and P. Sagaut, "Building efficient response surfaces of aerodynamic functions with Kriging and cokriging," *AIAA J.*, vol. 46, no. 2, pp. 498–507, Feb. 2008.

[33] M. A. Bouhlel and J. R. R. A. Martins, "Gradient-enhanced Kriging for high-dimensional problems," *Eng. Comput.*, vol. 35, no. 1, pp. 157–173, Jan. 2019.

[34] L. Chen, H. Qiu, L. Gao, C. Jiang, and Z. Yang, "Optimization of expensive black-box problems via gradient-enhanced kriging," *Comput. Methods Appl. Mech. Eng.*, vol. 362, Apr. 2020, Art. no. 112861.

[35] L. Laurent, R. Le Riche, B. Soulier, and P.-A. Boucard, "An overview of gradient-enhanced metamodels with applications," *Arch. Comput. Methods Eng.*, vol. 26, no. 1, pp. 61–106, Jan. 2019.

[36] G. Raggi, I. F. Galván, C. L. Ritterhoff, M. Vacher, and R. Lindh, "Restricted-variance molecular geometry optimization based on gradient-enhanced kriging," *J. Chem. Theory Comput.*, vol. 16, no. 6, pp. 3989–4001, May 2020.

[37] F. Cao, Z. Tang, C. Zhu, and X. Zhao, "An efficient hybrid multi-objective optimization method coupling global evolutionary and local gradient searches for solving aerodynamic optimization problems," *Mathematics*, vol. 11, no. 18, p. 3844, Sep. 2023.

[38] Z.-H. Han and S. Görtz, "Hierarchical Kriging model for variable-fidelity surrogate modeling," *AIAA J.*, vol. 50, no. 9, pp. 1885–1896, Sep. 2012.

[39] Z.-H. Han, S. Görtz, and R. Zimmermann, "Improving variable-fidelity surrogate modeling via gradient-enhanced Kriging and a generalized hybrid bridge function," *Aerosp. Sci. Technol.*, vol. 25, no. 1, pp. 177–189, Mar. 2013.

[40] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Global Optim.*, vol. 13, no. 4, pp. 455–492, Dec. 1998.

[41] M. Kanazaki, T. Matsuno, K. Maeda, and H. Kawazoe, "Efficient global optimization applied to wind tunnel evaluation-based optimization for improvement of flow control by plasma actuators," *Eng. Optim.*, vol. 47, no. 9, pp. 1226–1242, Sep. 2014.

[42] S. Xu, H. Chen, and J. Zhang, "A study of Nash-EGO algorithm for aerodynamic shape design optimizations," *Structural Multidisciplinary Optim.*, vol. 59, no. 4, pp. 1241–1254, Apr. 2019.

[43] D. Ginsbourger, R. Le Riche, and L. Carraro, "Kriging is well-suited to parallelize optimization," in *Computational Intelligence in Expensive Optimization Problems* (Adaptation Learning and Optimization), vol. 2, Y. Tenne and C. K. Goh, Eds., Berlin, Germany: Springer, 2010, pp. 131–162.

[44] D. Zhan, J. Qian, and Y. Cheng, "Pseudo expected improvement criterion for parallel EGO algorithm," *J. Global Optim.*, vol. 68, no. 3, pp. 641–662, Jul. 2017.

[45] B. Vuillod, M. Zani, L. Hallo, and M. Montemurro, "Handling noise and overfitting in surrogate models based on non-uniform rational basis spline entities," *Comput. Methods Appl. Mech. Eng.*, vol. 425, May 2024, Art. no. 116913.

[46] Q. Wang, T. Nakashima, C. Lai, X. Du, T. Kanehira, Y. Konishi, H. Okuizumi, and H. Mutsuda, "An improved system for efficient shape optimization of vehicle aerodynamics with 'noisy' computations," *Structural Multidisciplinary Optim.*, vol. 65, no. 8, Aug. 2022.

[47] J. Ollar, C. Mortished, R. Jones, J. Sienz, and V. Toropov, "Gradient based hyper-parameter optimisation for well conditioned Kriging metamodels," *Struct. Multidisciplinary Optim.*, vol. 55, no. 6, pp. 2029–2044, Jun. 2017.

[48] Q. Wang, T. Nakashima, C. Lai, B. Hu, X. Du, Z. Fu, T. Kanehira, Y. Konishi, H. Okuizumi, and H. Mutsuda, "Enhanced expected hypervolume improvement criterion for parallel multi-objective optimization," *J. Comput. Sci.*, vol. 65, Nov. 2022, Art. no. 101903.

[49] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[50] F. Vandenbergh and A. Engelbrecht, "A study of particle swarm optimization particle trajectories," *Inf. Sci.*, vol. 176, no. 8, pp. 937–971, Apr. 2006.

[51] J. Liang, B. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," Dept. Comput. Intell. Lab., Zhengzhou Univ., Nanyang Technological University,, Zhengzhou, China, Tech. Jan. 2013, vol. 201212, no. 34, pp. 281–295.

[52] S. Das and P. N. Suganthan, "Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems," Nanyang Technological Univ., Singapore, Tech. Rep., 2010, pp. 341–359.

[53] A. Grosso, A. R. M. J. U. Jamali, and M. Locatelli, "Finding maximin Latin hypercube designs by iterated local search heuristics," *Eur. J. Oper. Res.*, vol. 197, no. 2, pp. 541–547, Sep. 2009.

[54] Q. Wang, T. Nakashima, T. Kanehira, and H. Mutsuda, "A Python toolbox for surrogate-based optimization," in *Proc. IEEE 12th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Beijing, China, Jul. 2022, pp. 80–84.

[55] D. W. Zimmerman and B. D. Zumbo, "Relative power of the Wilcoxon test, the Friedman test, and repeated-measures ANOVA on ranks," *J. Exp. Educ.*, vol. 62, no. 1, pp. 75–86, Jul. 1993.

[56] J. Kou, L. Botero-Bolívar, R. Ballano, O. Marino, L. de Santana, E. Valero, and E. Ferrer, "Aeroacoustic airfoil shape optimization enhanced by autoencoders," *Expert Syst. Appl.*, vol. 217, May 2023, Art. no. 119513.

[57] Z. Chen, X. Wei, T. Xiao, and N. Qin, "Optimization of transonic low-Reynolds number airfoil based on genetic algorithm," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 238, no. 1, pp. 44–60, Nov. 2023.

[58] Y. Yu, Z. Lyu, Z. Xu, and J. R. R. A. Martins, "On the influence of optimization algorithm and initial design on wing aerodynamic shape optimization," *Aerosp. Sci. Technol.*, vol. 75, pp. 183–199, Apr. 2018.

[59] J. L. Thomas and M. D. Salas, "Far-field boundary conditions for transonic lifting solutions to the Euler equations," *AIAA J.*, vol. 24, no. 7, pp. 1074–1080, 1986.

[60] R. F. Soares, K. P. Garry, and J. Holt, "Comparison of the far-field aerodynamic wake development for three DrivAer model configurations using a cost-effective RANS simulation," SAE Technical Paper 2017-01-1514, Mar. 2017.

[61] *NASA Langley Turbulence Modeling Resource Website*. [Online]. Available: http://turbmodels.larc.nasa.gov.

[62] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Jan. 2020.

[63] M. Sato, Y. Kodama, M. Tsuji, and T. Odajima, "Co-design and system for the supercomputer 'Fugaku,'" *IEEE Micro*, vol. 42, no. 2, pp. 26–34, Mar. 2022.

[64] N. Jansson, R. Bale, K. Onishi, and M. Tsubokura, "CUBE: A scalable framework for large-scale industrial simulations," *Int. J. High Perform. Comput. Appl.*, vol. 33, no. 4, pp. 678–698, Dec. 2018.

[65] I. S. Kavvadias, E. M. Papoutsis-Kiachagias, G. Dimitrakopoulos, and K. C. Giannakoglou, "The continuous adjoint approach to the k–ωSSTturbulence model with applications in shape optimization," *Eng. Optim.*, vol. 47, no. 11, pp. 1523–1542, Dec. 2014.



HANG FU is currently pursuing the Ph.D. degree in computing science with the Graduate School of System Informatics, Kobe University, Japan. He is also supported by China Scholarship Council (CSC) to focus on the establishment of a powerful optimization system for aerodynamic shape optimization. His main research interests include optimization of aerodynamic shape optimization and the investigation of the application of evolutionary algorithms in engineering optimization problems.



QINGYU WANG received the D.Eng. degree from Hiroshima University, Japan, in 2022. He is currently an Engineer with the Automobile Aerodynamic Optimization Design Department, Institute of Auto Engineering, BYD Auto Industry Company Ltd. His research interests include computational fluid dynamics, optimization of aerodynamic shape optimization, and application of evolutionary algorithms in engineering optimization problems.



TAKUJI NAKASHIMA was born in Osaka, Japan, in 1977. He received the D.Eng. degree from The University of Tokyo, in 2006. He is currently an Associate Professor with the Graduate School of Advanced Science and Engineering, Hiroshima University, Japan. He has a strong interest in data science and applied mathematics regarding the CFD with high-performance computing technology, such as algorithms for optimization and graph partitioning. His main research interests include applied aerodynamics using both computational fluid dynamics (CFD) and experimental fluid dynamics (EFD) in automobile, and shipbuilding engineering.



ASAHI KAWASAKI is currently pursuing the master's degree in computing science with the Graduate School of System Informatics, Kobe University, Japan. His main research interests include the multi-objective optimization of automobile aerodynamics and the investigation of the development and application of surrogate models and neural network models in automobile aerodynamic shape optimization.



CHENGUANG LAI received the Ph.D. degree from Tohoku University. He is currently a Professor with the College of Vehicle Engineering, Chongqing University of Technology, China. He is also a Collaborative Researcher and a Visiting Professor with Tohoku University. His main research interests include intelligent optimization and analysis for vehicle aerodynamics, vehicle body design, and body cooperative optimization.



KEIGO SHIMIZU received the Ph.D. degree from Hiroshima University. He is currently a Senior Research Engineer with the Research and Development Center, Mazda Motor Corporation, Japan. His main research interests include human comfort in automobiles and fluid dynamics in automobile engineering, such as vehicle aerodynamics and automotive air conditioning.



RAHUL BALE received the Ph.D. degree from Northwestern University, Evanston, IL, USA, in 2013. He is currently an Associate Professor with the Graduate School of System Informatics, Kobe University, Kobe, Japan. He is also a Research Scientist with the Complex Phenomena Unified Simulation Research Team, RIKEN Center for Computational Science, Kobe. His research interests include computational fluid dynamics, fluid-structure interaction, urban climate modeling, and high-performance computing. He received the ACM Gordan Bell Special Prize for his contributions to HPC CFD software development for COVID-related research, in 2021.



MAKOTO TSUBOKURA was born in Japan, in 1969. He received the Ph.D. degree from The University of Tokyo, in 1997. He is currently a Professor with the Graduate School of System Informatics, Kobe University, Kobe, Japan. He is also the Team Leader of the Complex Phenomena Unified Simulation Research Team, RIKEN, Advanced Institute for Computational Science, Kobe. He has been involved in developing CFD software for industrial applications for these 15 years. The CFD software has been developed on the supercomputer 'Fugaku' in Japan as a part of the national project. His main research interests include vehicle aerodynamics, sports aerodynamics, computational fluid dynamics, and massively parallel computing. He is a fellow of The Japan Society of Mechanical Engineers, The Japan Society of Fluid Mechanics, and The Japan Society of Automotive Engineers. He was awarded the ACM Gordon Bell Special Prize for High Performance Computing-Based COVID-19 Research, in 2021.