



OPEN ACCESS

EDITED BY

Mojtaba Nedaei,
University of Padua, Italy

REVIEWED BY

Bingyuan Hong,
Zhejiang Ocean University, China
Najabat Ali,
Hamdard University Islamabad, Pakistan
Mohsen Rashki,
University of Sistan and Baluchestan, Iran

*CORRESPONDENCE

Yong Chen,
✉ chenyong_101@csnu.edu.cn

RECEIVED 13 June 2023

ACCEPTED 08 August 2023

PUBLISHED 14 September 2023

CITATION

Chen Y, Wang L and Huang H (2023), An effective surrogate model assisted algorithm for multi-objective optimization: application to wind farm layout design.
Front. Energy Res. 11:1239332.
doi: 10.3389/fenrg.2023.1239332

COPYRIGHT

© 2023 Chen, Wang and Huang. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

An effective surrogate model assisted algorithm for multi-objective optimization: application to wind farm layout design

Yong Chen^{1*}, Li Wang² and Hui Huang³

¹Innovation and Entrepreneurship College, Changsha Normal University, Changsha, China, ²School of Automation, Central South University, Changsha, China, ³School of Physical Education, Huanan Normal University, Changsha, China

Due to the intricate and diverse nature of industrial systems, traditional optimization algorithms require a significant amount of time to search for the optimal solution throughout the entire design space, making them unsuitable for meeting practical industrial demands. To address this issue, we propose a novel approach that combines surrogate models with optimization algorithms. Firstly, we introduce the Sparse Gaussian Process regression (SGP) into the surrogate model, proposing the SGP surrogate-assisted optimization method. This approach effectively overcomes the computational expense caused by the large amount of data required in Gaussian Process model. Secondly, we use grid partitioning to divide the optimization problem into multiple regions, and utilize the multi-objective particle swarm optimization algorithm to optimize particles in each region. By combining the advantages of grid partitioning and particle swarm optimization, which overcome the limitations of traditional optimization algorithms in handling multi-objective problems. Lastly, the effectiveness and robustness of the proposed method are verified through three types of 12 test functions and a wind farm layout optimization case study. The results show that the combination of meshing and SGP surrogate enables more accurate identification of optimal solutions, thereby improving the accuracy and speed of the optimization results. Additionally, the method demonstrates its applicability to a variety of complex multi-objective optimization problems.

KEYWORDS

multi-objective optimization, sparse Gaussian process, surrogate model, adaptive grid multi-objective particle swarm optimization algorithm, wind power engineering

1 Introduction

With the continuous advancement of technology, optimization problems have been widely applied in various fields, including industrial manufacturing, energy management, finance, and transportation. The primary objective of research on optimization problems is to achieve the optimal performance of a specific system or process. Generally, optimization problems can be divided into static optimization problems and dynamic optimization problems (Mavrovouniotis et al., 2017). Static optimization problems usually involve determining certain input conditions to maximize or minimize specific goals. Dynamic

optimization problems consider how to adjust control parameters or certain input conditions to achieve the best effect as time progresses. However, as the scale of industrial systems continues to expand, the complexity of controlled processes also continues to rise (Vafadar et al., 2021). Consequently, optimization problems have become more challenging and intricate. In some cases, complex systems are often represented using high-precision models. To solve these problems, complex systems are often described using high-precision models. However, this approach requires a substantial investment of time and manpower, leading to a lack of efficient methods for solving optimization problems. As a result, there is a growing emphasis on discovering faster and more effective optimization techniques to enhance both efficiency and accuracy.

The multi-objective particle swarm optimization (MOPSO) algorithm plays a significant role in addressing complex multi-objective optimization problems (Parsopoulos and Vrahatis, 2008). By optimizing the position of each particle, the algorithm searches for the optimal solution set and offers several advantages: Firstly, it efficiently identifies and outputs the Pareto optimal solution set, providing more decision-making options compared to other common multi-objective optimization algorithms. Secondly, the algorithm can simultaneously handle multiple-objective functions by adjusting weights to balance the priorities between objectives, a swiftly computing the optimal solution for multiple objectives. Lastly, the algorithm demonstrates strong performance in solving intricate optimization problems, including non-linearity and high dimensionality. MOPSO have proven successful in tackling real-world multi-objective problems, such as feature selection for medical diagnosis and other applications, optimization of stamping process parameters, optimization of indoor CO₂ and PM2.5 levels, ridesharing, concentration analysis, and optimization of manufacturing processes. However, when dealing with multi-objective problems, certain challenges arise, such as selecting the individual best solutions, identifying the global best solutions, and managing computational resources. For MOPSO, the first issue is addressed by randomly selecting one of the relatively superior particles in the space as the historical best. To solve the problem of global best solutions, MOPSO select the global best solution from the optimal set based on particle density, and preferably choose positions where particles are relatively rare in space as “elite particles” to lead other particles in searching for the optimal solution (Palmer, 2019). Furthermore, solving multi-variable optimization problems with MOPSO algorithms has increased the computational difficulty of the algorithm, leading to significant time and cost overhead. In recent years, optimization algorithms based on surrogate models have attracted increasing attention and are widely used due to their efficiency and applicability (Gu et al., 2021).

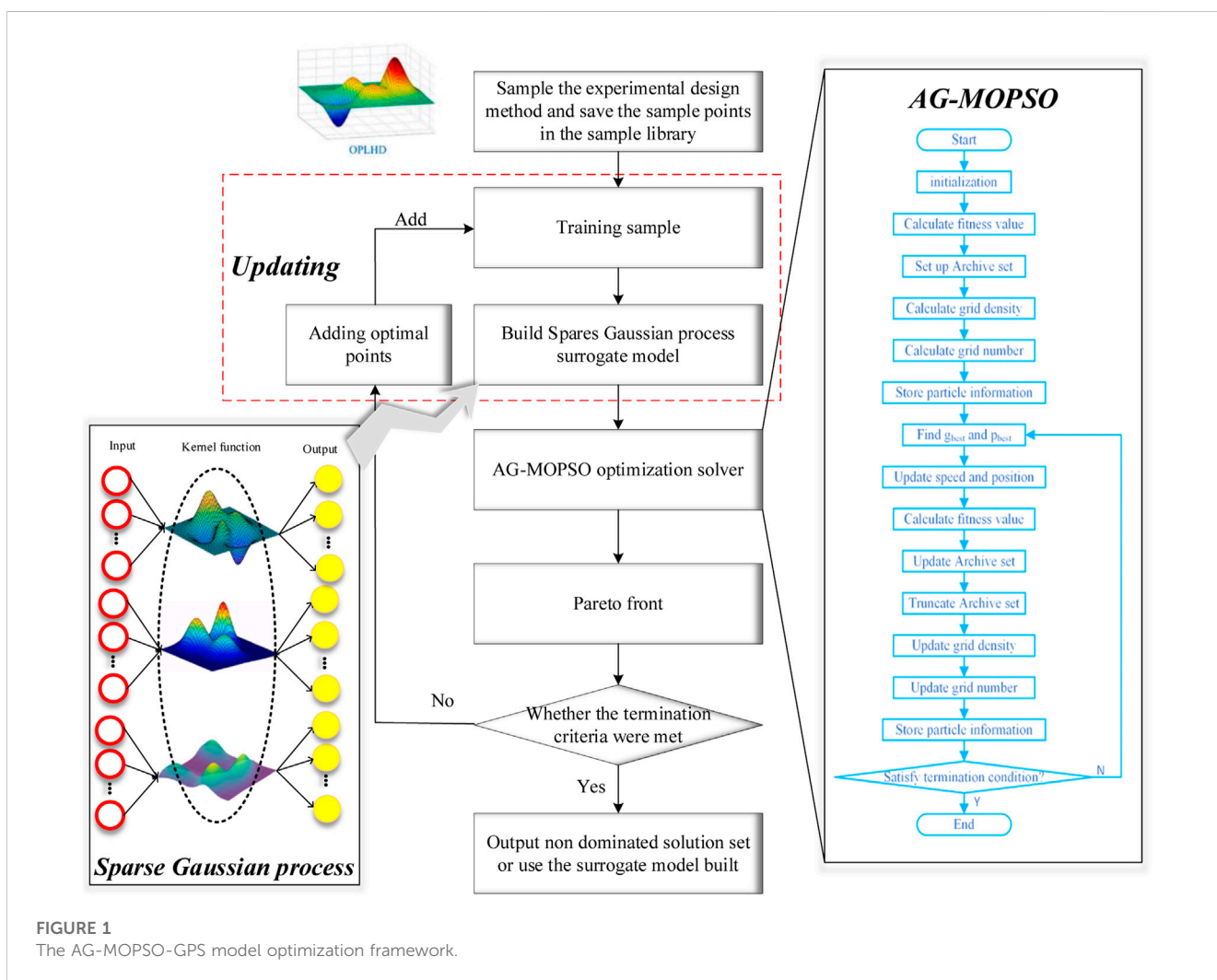
With the rapid development of computer testing and analysis methods, surrogate models have emerged as the primary approach for solving the black box problems, replacing time-consuming experiments. Surrogate model aided optimization algorithms can be divided into two categories: global surrogate model and local surrogate model (Zhou et al., 2006; Han and Zhang, 2012; Liu et al., 2016). The global surrogate model is employed to approximate the overall behavior of the problem and capture the general search direction, such as response surface methodology (RSM) (Wang Z. et al., 2022), Gaussian process (GP) (Shadab et al., 2022), radial basis function (RBF) model (Chen et al., 2022) and spline method

(SM) (Grimstad et al., 2016). On the other hand, the local surrogate model is utilized to enhance the approximation accuracy within a confined search space, such as Kriging model (Jeong et al., 2005), locally weighted regression (Talgorn et al., 2018), and support vector machine regression (SVM) (Cicczazzo et al., 2015), etc. Compared to methods like artificial neural network (ANN) models, surrogate models based on the Bayesian framework have been widely adopted due to their high model fitting accuracy and the ability to provide approximate values of uncertainty, which proves highly effective in model management (Lystad et al., 2023). Among these, surrogate optimization based on the Kriging model and GP model are typical examples of Bayesian-based surrogate optimization methods (Liu J. et al., 2023). The Kriging surrogate model, initially used in geology, has evolved into an optimization design method applicable to various disciplines (Currin et al., 1991; Kudela and Matousek, 2022). Jones constructed the most widely used improvement expectation criterion in the optimization field by using the Kriging model to predict the function estimation and its mean square error (Jones et al., 1998). Additionally, Jones proposed a highly efficient global optimization algorithm called Efficient Global Optimization, which maximizes the expected improvement criterion. Subsequently, researchers further improved the model and introduced methods such as Blind Kriging (Joseph et al., 2008), Co-Kriging (Liu et al., 2022), and Gradient Kriging (Liu F. et al., 2023), thereby expanding the selection range of Kriging models in surrogate modeling. Currently, the Kriging surrogate model has been widely applied in fields such as uncertainty analysis, reliability assessment, and optimization (Wang C. et al., 2022; Ling et al., 2022; Zhao et al., 2022). However, these methods have significant requirements for the initial data, making them unsuitable for situations with a limited number of samples or incomplete information during the early stages of product or equipment development. In addition, the GP model is considered a surrogate model with low modeling complexity that balances accuracy and correlation. It is extensively applied in complex structures, particularly those with multiple failure modes, and situations where reliability calculations are time-consuming (Liu et al., 2013; Satria Palar et al., 2020). Many researchers have conducted in-depth research and exploration on the GP surrogate model, expanding its application fields and making it one of the most commonly used optimization methods. Su et al. (2017) proposed a dynamic GPR surrogate model based on Monte Carlo simulation for the reliability analysis of bridge engineering structures. Avendaño-Valencia et al. (2021) proposed a GP surrogate model method based on Bayesian hyperparameter calibration to predict short-term fatigue damage equivalent loads of wind turbines, achieving a prediction error of less than 4%. Golparvar et al. (2021) examined the impact of wind wave-related factors on the variability of offshore wind power using a GP surrogate model, and revealed the potential relationship between controlling offshore weather and power conversion. Furthermore, Preen et al. (2019) simulated tumor growth using a GP surrogate model, opening up broad prospects for dealing with complex, multiscale tumor models.

In summary, Kriging surrogate model and GP surrogate model are widely applied due to their excellent performance in high-precision fitting, optimization, and uncertainty analysis. They

TABLE 1 Advantages and disadvantages of different surrogate model.

Surrogate model	Advantage	Disadvantage
Kriging (Jeong et al., 2005)	<ul style="list-style-type: none"> • Effective handling of non-linear problems and number of noises 	<ul style="list-style-type: none"> • Requires a lot of data • Takes a long time to compute • Complex for high-dimensional problems
RBF (Chen et al., 2022)	<ul style="list-style-type: none"> • Strong data adaptability, low noise sensitivity, fast calculation speed etc. 	<ul style="list-style-type: none"> • Appropriate kernel functions and parameters need to be selected artificially • Large calculation error when the number of input data is large
SVM (Cicczazo et al., 2015)	<ul style="list-style-type: none"> • For problems of high-dimensional data classification with good effect • No need to understand data characteristics 	<ul style="list-style-type: none"> • Sensitive to noise data • High modeling complexity
ANN (Liu et al., 2021)	<ul style="list-style-type: none"> • Ability to deal with non-linear problems 	<ul style="list-style-type: none"> • A large amount of data is required to train the model • The model structure is complex • The interpretation of data is weak



offer effective global optimization, interpretability, and flexibility, making them versatile choices in various domains. However, both Kriging surrogate model and GP surrogate model require complex

computations to estimate functions and establish correlations. Dealing with large input spaces or extensive datasets significantly increases the computational cost. Furthermore, the fitting accuracy

TABLE 2 Comparative analysis of IGB measures for test functions.

Test function	Statistical value	AG-MOPSO-GPS	AG-MOPSO	NSGA-II	AG-MOPSO-ANNS	AG-MOPSO-SVMS	
ZDT1	mean	0.0023	0.068	0.072	0.0043	0.0051	
	std	2.173E-4	3.365E-4	5.782E-4	2.178E-4	4.327E-4	
ZDT2	mean	0.0037	0.093	0.067	0.0058	0.0061	
	std	2.316E-3	3.175E-3	4.247E-3	2.768E-3	2.948E-3	
ZDT3	mean	0.0643	0.0954	0.1084	0.0616	0.0713	
	std	0.0129	0.0201	0.0374	0.0193	0.0226	
DTLZ1	mean	0.0479	0.0632	0.0912	0.0501	0.0578	
	std	4.679E-3	4.721E-3	5.156E-3	4.956E-3	4.785E-3	
DTLZ2	mean	0.0565	0.0734	0.0638	0.0593	0.0721	
	std	3.158E-3	5.123E-3	4.148E-3	9.148E-2	4.256E-3	
DTLZ7	mean	0.1930	0.3219	0.2753	0.2004	0.2314	
	std	0.0133	0.0262	0.0194	0.0158	0.0167	
3	WFG1	mean	1.7483	2.7980	2.3490	1.8436	1.7738
		std	0.0943	0.1573	0.0754	0.0519	0.0930
	WFG2	mean	0.8573	0.9357	1.0753	0.8754	0.8932
		std	0.1329	0.0873	0.1420	0.2243	0.1954
	WFG3	mean	0.2107	0.3113	0.3721	0.2843	0.2585
		std	0.0378	0.0539	0.0493	0.0447	0.0461
4	WFG1	mean	1.8391	2.4394	2.5730	1.9532	2.0967
		std	0.0573	0.0625	0.0843	0.0714	0.0684
	WFG2	mean	1.2342	1.5843	1.7922	1.3075	1.5493
		std	0.1454	0.2256	0.1739	0.2168	0.1633
	WFG3	mean	0.3745	0.4593	0.4303	0.3981	0.3825
		std	0.0384	0.0534	0.0450	0.0487	0.5512

Bold values represents the optimal results.

of the model is greatly affected when there is a small initial dataset. Additionally, different surrogate models and optimization algorithms have varying advantages and disadvantages, which necessitate the use of other evaluation methods to measure their performance and applicability. Therefore, it is essential to compare and analyze optimization algorithms based on surrogate models. In this paper, different optimization algorithms of surrogate models are compared in order to better understand the surrogate models. The specific contributions of this article are as follows.

1. This paper systematically summarizes the deficiencies of traditional proxy model optimization, particularly the adaptability issue between surrogate models and optimization problems. The paper combines the Sparse Gaussian Process (SGP) surrogate method with optimization algorithms for the first time. By employing the SGP as the surrogate model, it can estimate unknown function values a small amount of known data and provide accurate uncertainty information.

2. This paper introduces a novel surrogate model method called the Adaptive Grid-based Multi-Objective Particle Swarm Optimization algorithm (AG-MOPSO), which is based on SGP surrogate modeling. The use of an adaptive grid divides the optimization problem into multiple regions and utilizes the MOPSO algorithm to optimize particles in each region, addressing the inefficiency problems of traditional algorithms in complex scenarios.

3. To validate the proposed method, three sets of test functions and a wind farm layout optimization case study are employed for testing. Comparative analysis with different optimization algorithm models demonstrates the superior performance of the proposed method in terms of accuracy and optimization speed, particularly for multi-objective optimization problems and diverse, complex industrial systems, when compared to alternative optimization algorithms.

The remaining sections of this paper are organized as follows: Section 2 introduces multi-objective optimization problems and

TABLE 3 Comparative analysis of HV measures for test functions.

Test function	Statistical value	AG-MOPSO-GPS	AG-MOPSO	NSGA-II	AG-MOPSO-ANNS	AG-MOPSO-SVMS	
ZDT1	mean	121.5421	109.5395	108.3452	121.1453	120.7460	
	std	0.0926	0.0345	0.7042	0.1325	0.7321	
ZDT2	mean	121.9517	110.1345	109.4098	121.8732	121.0545	
	std	0.4739	0.5017	0.7931	0.6333	0.6234	
ZDT3	mean	115.4395	109.6390	109.2930	114.7145	115.0724	
	std	1.5475	1.9560	0.8654	1.2365	0.9653	
DTLZ1	mean	14.5612	9.2365	7.2350	13.5353	13.1207	
	std	0.1172	0.1289	0.1731	0.2352	0.4521	
DTLZ2	mean	14.2974	8.7402	9.1264	13.3467	11.4579	
	std	0.0547	0.3589	0.4524	0.2938	0.0327	
DTLZ7	mean	39,543.1	39,432.8	39,312.0	39,473.5	39,502.9	
	std	19.4343	20.5839	21.5829	19.9505	20.1587	
3	WFG1	mean	114.5409	111.0384	112.5791	114.0759	114.2306
		std	0.0137	0.0661	0.0587	0.0216	0.0357
	WFG2	mean	208.5743	200.5863	201.4730	205.4918	206.5783
		std	2.4323	1.9684	1.6043	3.5333	1.6733
	WFG3	mean	160.4532	140.5830	149.3402	150.5945	155.6320
		std	2.5303	1.6493	2.5835	1.2010	1.5294
4	WFG1	mean	743.8567	721.8540	718.5603	735.3200	738.0541
		std	13.4005	15.3945	18.4839	19.5584	15.5395
	WFG2	mean	972.4356	956.4309	890.5437	969.4332	985.1439
		std	45.3892	53.8875	50.4820	49.6843	50.3205
	WFG3	mean	802.5640	765.4302	772.6403	795.5424	790.2093
		std	13.0984	12.5834	11.5473	14.2034	13.6803

Bold values represents the optimal results.

compares the advantages and disadvantages of commonly used surrogate model-assisted optimizations. Section 3 provides a detailed description of the implementation steps of the adaptive grid multi-objective particle swarm optimization algorithm based on the SGP surrogate (AG-MOPSO-GPS) model. Section 4 evaluates the proposed surrogate model-assisted optimization algorithm. Section 5, the algorithm proposed in this paper is applied to actual engineering case. Finally, section 6 summarizes the contributions of this paper.

2 Optimization algorithm and surrogate model

2.1 Multi-objective optimization problem

Multi-objective optimization is a prevalent challenge in modern complex industrial systems (Cui et al., 2017). Achieving

simultaneous optimization for each objective, given their conflicting nature, is often unattainable. In general, multi-objective optimization problems are composed of several objective functions, some related equations and inequality constraints. Mathematically, they can be described as follows:

$$\min F(X) = [f_1(x), f_2(x), \dots, f_m(x)]^T \tag{1}$$

$$\begin{aligned} s.t. \quad & g_i(x) \leq 0, i = 1, 2, 3, \dots, p \\ & h_j(x) \leq 0, i = 1, 2, 3, \dots, p \end{aligned} \tag{2}$$

Where, $f_i(x), \{i = 1, 2, 3, \dots, m\}$ represents the objective function; $g_i(x)$ and $h_j(x)$ are the constrained function; $x = \{x_1, x_2, \dots, x_n\}$ is an n-dimensional design variable. $X = \{x|x \in R^n, g_i(x) \geq 0, h_j(x) = 0, i = 1, 2, \dots, p, j = 1, 2, \dots, q\}$ represents the feasible range of the objective function.

In multi-objective optimization, a significant challenge arises due to the mutual constraints between various objectives. This constraint often leads to a trade-off where the improvement in

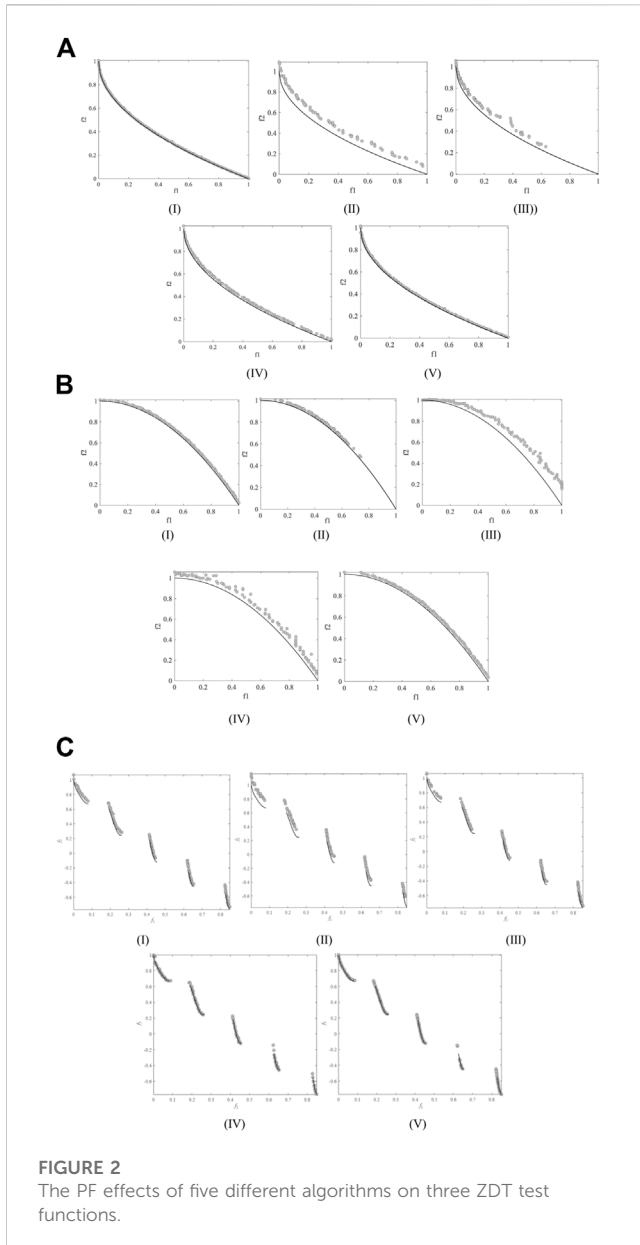


FIGURE 2
The PF effects of five different algorithms on three ZDT test functions.

one target's performance comes at the cost of sacrificing the performance of other targets. As a result, finding a single optimal solution becomes impractical in most cases. To address this issue, multi-objective optimization problems typically use a set of non-dominated solutions to represent all optimal solutions. This set of solutions is called the Pareto optimal solution set. Finding the Pareto optimal solution set is a very important task in multi-objective optimization problems. One way to solve this task is to find as many Pareto optimal solutions as possible for the optimization problem. This means finding a set of solutions that are as close to the Pareto optimal domain as possible, and finding a set of solutions that are as different as possible. So, this problem can usually be solved through multiple relatively balanced solutions, rather than a single optimal solution. The Pareto optimal solution set is a typical solution in multi-objective optimization problems. In fact, the Pareto optimal solution set is a model that can represent future needs and user decisions, as

most users pursue maximum benefits without losing any purpose. Meanwhile, solving Pareto optimality problems can help people better understand and apply multi-objective optimization problems. In summary, the Pareto optimal solution set in multi-objective optimization problems is an important form of solving the optimal solution. By finding a set of Pareto optimal solutions, people can obtain multiple equilibrium solutions to solve the problem. The Pareto optimal solution set represents a feasible solution that can help people better apply multi-objective optimization problems to practical applications. Currently, the majority of evolutionary algorithms used to address multi-objective optimization problems often require tens of thousands of fitness evaluations, incurring significant expenses in terms of time and resources. Despite numerous research efforts to create novel optimization algorithms for solving multi-objective optimization problems, the trade-off between solution speed and the number of function evaluations is rarely considered (Akinola et al., 2022). As a solution to mitigate costly function evaluations, data-driven surrogate-assisted evolutionary algorithms have emerged. This method involves constructing a surrogate model to approximate the original expensive objective function and then employing the surrogate model to evaluate a subset of candidate solutions, thereby reducing computational costs (He et al., 2023). It has become a prevalent approach to alleviate the burden of costly function evaluations in multi-objective optimization problems.

2.2 Surrogate model

The surrogate model method offers a solution to the optimization design and design analysis challenges posed by complex source models. It involves constructing substitute models for these source models. Despite requiring the expensive valuation of source models during the sampling process, this method effectively reduces the number of simulations in optimization, thereby improving its efficiency. Common surrogate models include the Kriging model, RBF model, ANN model, and others. In practice applications, these surrogate models are often combined with various optimization algorithms, such as the genetic algorithm, particle swarm algorithm, simulated annealing, among others (Nguyen et al., 2014). A comparison of several classic surrogate models is presented in Table 1. The Kriging surrogate model exhibits high fitting accuracy and can provide predicted and error values. However, constructing the Kriging model can be time-consuming, especially when dealing with numerous sample points. In contrast, the RBF surrogate model is relatively easy to construct, flexible, and adaptable to nonlinear problems. The difficulty of constructing the SVM surrogate model is determined by the number of support vectors requires. Although it can reduce the problem of "curse of dimensionality" to some extent, it cannot handle large sample sets. Meanwhile, the ANN surrogate model is widely used due to its effective learning mechanism, accuracy, and robustness, although constructing a high-precision ANN surrogate model takes a lot of time (Liu et al., 2021).

Despite the strengths exhibited by different surrogate models, choosing the most appropriate model remains a challenging problem in practice. Multiple criteria need to be considered simultaneously, such as model accuracy, construction time, and computational efficiency. Moreover, in optimization scenarios

3 Surrogate assisted optimization framework

Utilizing surrogate models in optimization is a vital technique that can significantly save computational resources, especially for multi-objective optimization problems, where evaluating each objective function can be time-consuming. The main idea behind surrogate model-assisted optimization is to use a surrogate model to approximate the true objective function, thus speeding up the optimization process and improving its efficiency. As a result, it has been extensively studied. In this section, we address the adaptability deficiencies of existing surrogate models in optimization problems and propose the AG-MOPSO-GPS optimization algorithm. We will provide a detailed introduction to the SGP surrogate model, the AG-MOPSO Optimization algorithm, and the framework of surrogate model-assisted optimization.

3.1 SGP surrogate models

As a type of machine learning algorithm, the GP model integrates the advantages of traditional statistical theory and Bayesian theorem (Zhang et al., 2019). Compared to linear regression, the advantages of the GP model lie in its ability to solve complex problems with small samples, high dimensions, and non-linearity (Giovanis and Shields, 2020). Compared to black box algorithms such as artificial neural networks, its advantages include ease of implementation, adaptive hyperparameters, and interpretability (Ghahramani, 2015). The GP model allows prior distributions to be placed over the entire function for inference, rather than just learning model parameters. Consider a dataset $\mathcal{D} = \{(x^i, f(x^i))\}_{i=1}^{N_i}$ of N samples. The GP model is defined as a set of random variables $(x_1, f(x_2), \dots, f(x_i))$, with any finite-dimensional distribution being a joint normal distribution.

$$f(x) \sim GP(m(x), K(x, x')) \tag{3}$$

Where, $m(x)$ represents mean function, $K(x, x')$ is kernel function.

The goal of the GP model is to model data as the output of certain functions. However, in normal cases, the measured data contains noise. To model this noise, a Gaussian noise model is commonly used:

$$y = f(x) + \varepsilon \quad \varepsilon \sim N(0, \sigma_n^2) \tag{4}$$

The prior distribution of the training samples follows the distribution given below:

$$y \sim N(m(x), K + \sigma_n^2 I)$$

Where, I represents the n -order identity matrix.

In order to make predictions, it is important to evaluate the joint prior distribution between the training and testing data:

$$\begin{bmatrix} y \\ y^* \end{bmatrix} \sim N\left(\begin{bmatrix} m(x) \\ m(x^*) \end{bmatrix}, \begin{bmatrix} K_{xx} + \sigma_n^2 I & K_{xx^*} \\ K_{x^*x} & K_{x^*x} + \sigma_n^2 I \end{bmatrix}\right) \tag{5}$$

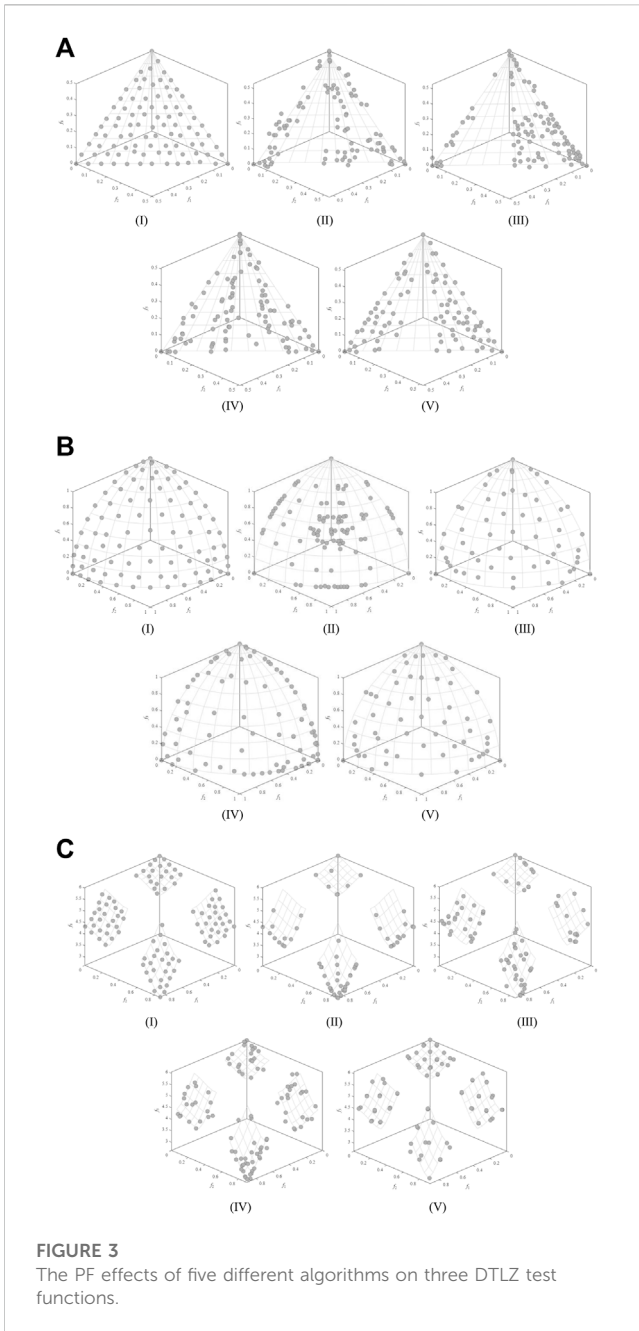


FIGURE 3
The PF effects of five different algorithms on three DTLZ test functions.

involving multiple objectives, the Pareto front (PF) of the surrogate model must be sufficiently sampled to accurately capture the entire set of solutions. In conclusion, the effective construction and updating of surrogate models are essential for the successful implementation of multi-objective evolutionary algorithms, which are central to many optimization problems. Advancements in this field necessitate a better understanding of the strengths and weaknesses of different surrogate models, along with the development of more efficient algorithms capable of accommodating the diverse requirements of various optimization scenarios.

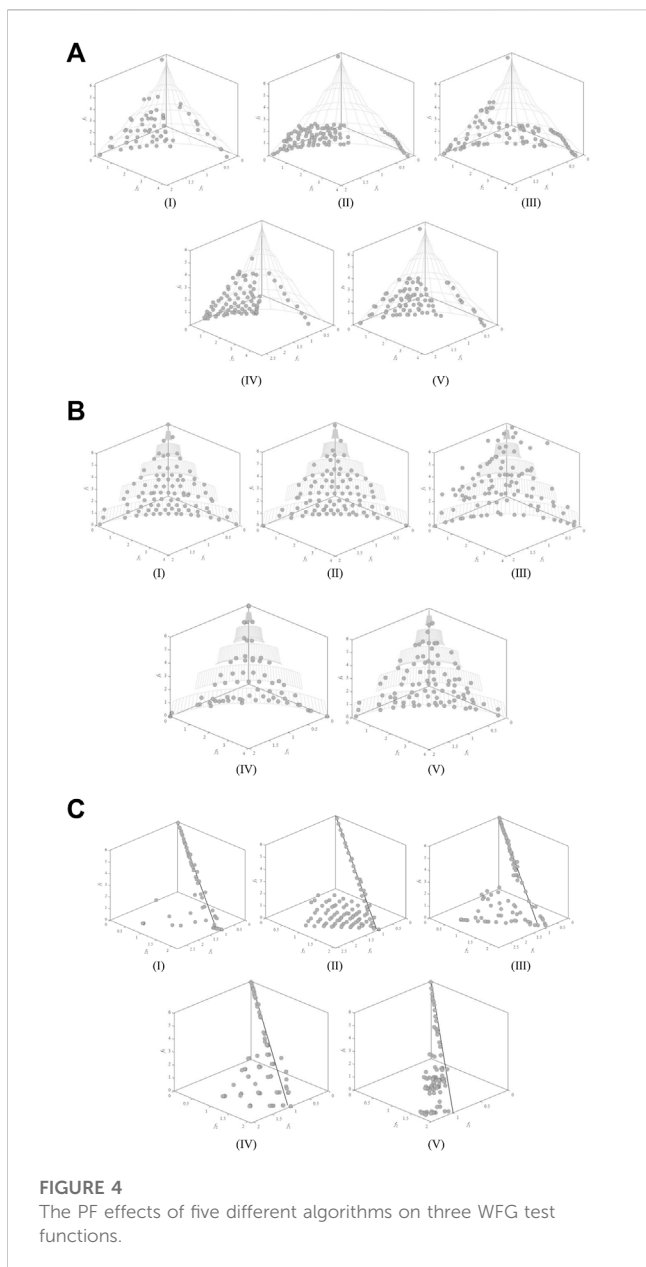


FIGURE 4
The PF effects of five different algorithms on three WFG test functions.

Where, x^* represents the new input dataset. y^* represents the predicted new potential unknown output. $K_{xx} = K(x, x)$, $K_{xx^*} = K(x, x^*)$, $K_{x^*x} = K(x^*, x)$, $K_{x^*x^*} = K(x^*, x^*)$.

By applying the properties of multivariate Gaussian distribution, it is known that each conditional distribution is also a Gaussian distribution. Utilizing this standard result, we can express the posterior distribution of y^* as:

$$\begin{aligned}
 p(y^* | x^*, x, y) &\sim N(\mu^*, \Sigma^*) \\
 \mu^* &= m(x^*) + K_{x^*x}(K_{xx} + \sigma_n^2 \mathbf{I})^{-1}(y - m(x)) \\
 \Sigma^* &= K_{x^*x^*} - K_{x^*x}(K_{xx} + \sigma_n^2 \mathbf{I})^{-1}K_{xx^*} + \sigma_n^2
 \end{aligned}
 \tag{6}$$

The kernel function is the most important component of the GP model, as it is used during the training phase to map the relationship between the input and output, and also for predicting new points through interpolation. The radial basis function kernel is commonly used, which has a small number of hyperparameters. These hyperparameters control the behavior of the kernel function, and in order to utilize the GP model to analyze the non-linear relationship between different features, these hyperparameters must be determined. Typically, the second type of maximum likelihood method is used to determine the hyperparameters of the kernel function. The objective of this method is to maximize the marginal likelihood (also known as model evidence) of the model. This optimization finds the minimum complex model using the Bayesian Occam’s razor, given the training set D. For convenience and numerical stability, this optimization is usually performed by minimizing the negative logarithm of the marginal likelihood. Therefore, the estimated value of the hyperparameters $\hat{\theta}$ is obtained through the following optimization.

$$\begin{aligned}
 \hat{\theta} &= \arg \min_{\theta} \{-\log p(y | x, \theta)\} \\
 -\log p(y | x, \theta) &= -\log N(m(x), K_{xx} + \sigma_n^2 \mathbf{I}) \\
 &= \frac{N}{2} \log(2\pi) + \frac{1}{2} \log |K_{xx} + \sigma_n^2 \mathbf{I}| \\
 &= \frac{1}{2} [(y - m(x))^T (K_{xx} + \sigma_n^2 \mathbf{I})^{-1} (y - m(x))]
 \end{aligned}
 \tag{7}$$

However, in the GP model, to determine hyperparameters or make predictions, it is necessary to evaluate the inverse of the covariance matrix with noise $(k_{xx} + \sigma_n^2 \mathbf{I})^{-1}$. This computation step carries a time complexity and memory storage of $O(N^3)$, which means that using the GP model to establish non-linear regression models for big datasets is not feasible. To address this issue, inducing data sets are constructed by randomly selecting $M (M \ll N)$ data points from the training set to learn the hyperparameters, and such models are known as SGPR. Introducing these inducing points reduces the complexity of the GP model from $O(N^3)$ to $O(MN^2)$. Utilizing the inducing points $\{Z, u\}$ (where Z

TABLE 4 Metrics performance for constant wind speed and variable wind speed, with $N_i = 15$.

	Statistical value	Constant wind speed (12 m/s)		Variable wind speed (8, 12, 17 m/s)	
		AG-MOPSO-GPS	MO-LSA	AG-MOPSO-GPS	NSGA-II
IGD	Mean	0.2945	0.4732	0.3831	0.5491
	Std	0.0057	0.0064	0.0074	0.0081
HV	Mean	1.3573	0.8437	0.9547	0.7458
	Std	0.0327	0.0364	0.2643	0.2722

Bold values represents the optimal results.

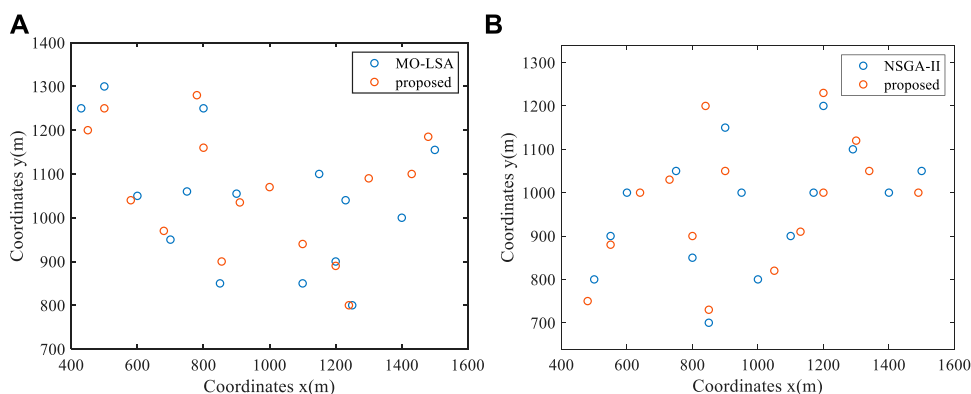


FIGURE 5 Layout with the best trade-off. [(A) constant wind speed, (B) variable wind speed].

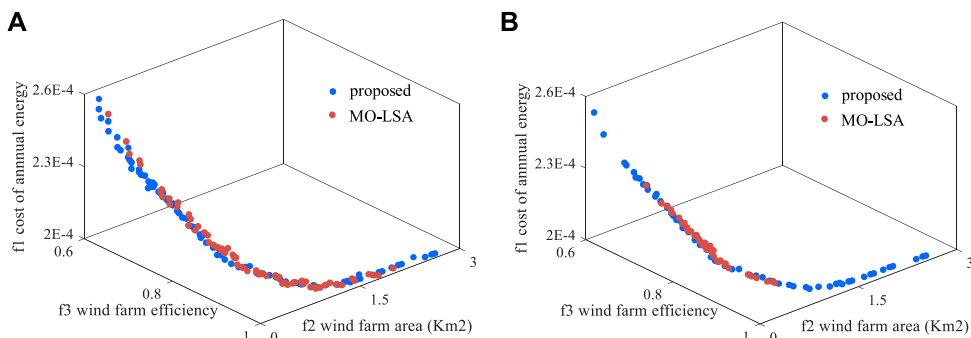


FIGURE 6 The PF of different optimization algorithms. [(A) constant wind speed, (B) variable wind speed].

represents the locations of inducing points and \mathbf{u} denotes the latent function values at these points) forms a variational approximation to the full posterior of the model. Then, the expected distribution value is obtained by minimizing the Kulback-Leibler divergence between the variational distribution and the posterior distribution:

$$q(\mathbf{y}^* | \mathbf{x}^*, \mathbf{x}, \mathbf{y}, \mathbf{u}) = N(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) \tag{8}$$

$$\boldsymbol{\mu}^* = Q_{x^*x} (Q_{xx} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y} \tag{9}$$

$$\boldsymbol{\Sigma}^* = K_{x^*x^*} - Q_{x^*x} (Q_{xx} + \sigma_n^2 \mathbf{I})^{-1} Q_{xx^*} \tag{10}$$

Where, Q_{xx} represents the approximate Covariance matrix, and the expression is as follows:

$$Q_{xx} = K_{xu} K_{uu}^{-1} K_{ux} \tag{11}$$

3.2 Adaptive grid multi-objective particle swarm optimization algorithm (AG-MOPSO)

Particle swarm optimization (PSO) is a heuristic algorithm inspired by the foraging behavior of birds. Due to its high convergence rate and simple search principle, PSO has been

widely used and developed in the field of multi-objective optimization (Coello et al., 2004). However, traditional MOPSO have some shortcomings, such as insufficient consideration of the selection of global optimal particles, the lack of pruning of the non-dominated set, and the imbalance between global and local search capabilities. Therefore, in this paper, an AG-MOPSO is selected to optimize multi-objective problems (Yang et al., 2008).

The AG-MOPSO algorithm adopts a dual-population approach, where one population is a standard PSO group, while the other population is used to store the non-dominated solutions discovered by the group after each iteration, known as the Archive set. The main part of AG-MOPSO is similar to the general PSO algorithm. It initializes a swarm of particles in an n-dimensional search space, where the movement of each particle depends on its position and velocity vector. The velocity and position vectors can be represented by the following equations:

$$\begin{aligned} \vec{X} &= [\vec{x}_1, \vec{x}_2, \vec{x}_3, \dots, \vec{x}_n] \\ \vec{V} &= [\vec{v}_1, \vec{v}_2, \vec{v}_3, \dots, \vec{v}_n] \end{aligned} \tag{12}$$

In the search space, particles adjust their position and speed vectors according to global searching experience g_{best} and individual local searching experience P_{best} with certain rules. In this way, global

TABLE 5 The optimal solutions of different optimization algorithms.

Statistical value	Constant wind speed (12 m/s)		Variable wind speed (8, 12,17 m/s)	
	AG-MOPSO-GPS	MO-LSA	AG-MOPSO-GPS	NSGA-II
f_1	2.037E-4	2.155E-4	1.974E-4	2.029E-4
f_2	0.231	0.257	0.269	0.28
f_3	0.9389	0.9114	0.9795	0.9641

searching ability and local searching ability can be included simultaneously. The update formula is:

$$\begin{aligned}
 v_i^n(k+1) &= \underbrace{\omega(k) \cdot v_i^n(k)}_{\text{previous velocity}} + \underbrace{\Lambda_1 \omega_1 (p_{best,i}^n - x_i^n(k))}_{\text{personal component}} \\
 &\quad + \underbrace{\Lambda_2 \omega_2 (g_{best,i}^n - x_i^n(k))}_{\text{global component}} \\
 x_i^n(k+1) &= x_i^n(k) + v_i^n(k+1)
 \end{aligned}
 \tag{13}$$

Where, $v_i^n(k+1)$ and $v_i^n(k)$ represent the i th particle in the n th dimension from the velocity vector of this iteration versus the last iteration, respectively. $x_i^n(k+1)$ and $x_i^n(k)$ represent the i th particle in the n th dimension the position vector from this iteration to the last iteration, respectively. $g_{best,i}^n$ and $p_{best,i}^n$ represent the position vectors of the global and historical best particles for the first particle i in the n th dimension, respectively. Λ_1 and Λ_2 are local and global learning factors, respectively. ω_1 and ω_2 are random numbers between $[0,1]$. $\omega(k)$ is the inertial factor given by the following formula:

$$\omega(k) = \omega_{max} - \frac{k}{k_{max}} \cdot (\omega_{max} - \omega_{min})
 \tag{14}$$

Where, k_{max} represents the total number of iterations. The variable inertia weight can guarantee the global search ability in the early search period and the local search ability in the late search period.

AG-MOPSO has three innovations: adaptive grid (AG) algorithm, global optimal particle selection, and Archive set truncation. Taking the two-dimensional target space optimization problem as an example, AG first calculates the search range of the target space after K iterations $[\min Y_1^k, \max Y_1^k]$ and $[\min Y_2^k, \max Y_2^k]$. Then the grid model is calculated according to the following formula:

$$\begin{aligned}
 \Delta Y_1^k &= (\max Y_1^k - \min Y_1^k) / M \\
 \Delta Y_2^k &= (\max Y_2^k - \min Y_2^k) / M
 \end{aligned}
 \tag{15}$$

Where, M represents dividing the target space into $M \times M$'s Grid, Y_1 and Y_2 are the values of the fitness function. In the early stage of evolution, M selects a smaller value to save computational costs. As the algorithm continues to iterate, M will adaptively increase to improve the resolution in the later stage of evolution, obtaining a more realistic optimal result. After obtaining the modulus of the grid, traverse the population to calculate the grid numbers of each particle. For particle i , its grid number is given by the following equation:

$$(x_1^i, x_2^i) = \left(\text{Int} \left(\frac{Y_1^i - \min Y_1^k}{\Delta Y_1^k} \right) + 1, \text{Int} \left(\frac{Y_2^i - \min Y_2^k}{\Delta Y_2^k} \right) + 1 \right)
 \tag{16}$$

Where Int means integer. After calculating the grid number of each particle, the grid density information of each grid can be obtained. After calculating the grid number of each particle, the

grid density information of each grid can be obtained. The selection of the globally optimal particles is based on the density information. Specifically, the lower the density of the grid in which the particles in the Archive are located, the greater the probability of being selected as g_{best} . The more particles in Archive concentration are better than the number of particles in the group, the greater the search potential of the particles. Based on this idea, the g_{best} corresponding to each particle in the group is determined using the following formula:

$$g_{best,j}^k = \left\{ i \mid \min \{ F(i), i \in S_j^k \} \right\}
 \tag{17}$$

Where, $F(i)$ indicates the number of particles in the grid in which i th particles reside. S_j^k indicates that the Archive set is better than the set of group particles j , which is determined by the following formula.

$$S_j^k = \{ i \mid \forall i \succ j \}
 \tag{18}$$

Where, \succ indicates the Pareto dominance. When there are more than one particle that meets the criteria, select the Archive set particle with the greatest potential as g_{best} :

$$g_{best,j}^k = \left\{ g_{best,j}^k \mid \max \{ i \mid i \prec j, i \in P_k \} \right\}
 \tag{19}$$

Where, P_k represents the population of particles in the k th iteration. This formula represents the particle that dominates the maximum number of particles in the population of all the particles satisfying formula (17). When there are particles with the same potential, choose one randomly. The Archive Set truncation technique used by AG-MOPSO is based on the density information of the mesh. When the density of the mesh exceeds the threshold, the smallest potential particles are deleted according to (19), which ensures that Archive Set always has excellent search potential throughout the iteration process.

In the AG-MOPSO algorithm, the grid quantity is adaptively adjusted based on the particle distribution and changes during the optimization process. The variation of grid quantity depends on the current particle distribution and grid density, aiming to better adapt to the characteristics and changes of the problem during the optimization process. In the initial stage of the algorithm, the grid quantity may be relatively small because there is not enough information to accurately partition the optimization space. As the optimization progresses and particles move and search, the optimization space is gradually divided into more grids. By adaptively adjusting the grid quantity, the AG-MOPSO algorithm can efficiently search for the global optimal solution or the Pareto optimal solution set in different optimization stages and features of the optimization space. This flexible variation of grid quantity allows the algorithm to better adapt to complex multi-

objective optimization problems and maintain good performance throughout the optimization process.

3.3 Optimization framework and process description

When dealing with expensive multi-objective optimization problems, it is important to consider the significant uncertainty in the approximation of individuals by surrogate models (Zheng et al., 2022). This means that the areas surrounding these solutions may not be effectively explored, resulting in a high probability of finding suboptimal solutions. Furthermore, evaluating these solutions and using them to update the surrogate model could be the most effective way to enhance the accuracy of the model. Bayesian theory is an effective tool for considering the uncertainty in optimization problems. It can predict the objective function value and improve the variance of this predicted value. The optimization framework assisted by surrogate models takes a similar approach to Bayesian optimization, integrating the construction and optimization processes of surrogate models. In this study, we used an optimization framework assisted by a GP surrogate model (Sun et al., 2017), whose pseudocode is shown in Algorithm 1. The framework combines outer and inner optimization, where the outer layer updates the training set from the solution output by the inner layer, iteratively updating the surrogate model, while the inner layer iteratively optimizes to obtain the function and then the sampling points to output to the outer layer.

An optimization framework assisted by GP surrogate models

Input: Relevant parameters: \mathcal{D} , iteration number $Iter_{max}$, Set the initial number of samples n

Output: Optimal sample set $\{f(x)\}$, current optimal solution $\min_{x \in \mathcal{D}} f(x)$

Initialization:

Using experimental design methods to initialize a solution set $\mathcal{X} \leftarrow \{x^i\}_{i=1}^{N_i}$ from Ω and evaluate their objective function values $\mathcal{Y} \leftarrow \{f(x^i)\}_{i=1}^{N_i}$. Set initial training set $\mathcal{D} \leftarrow \{(x^i, f(x^i))\}_{i=1}^{N_i}$. Set initial $Iter \leftarrow 0$

While $Iter < Iter_{max}$ do

Build a GP model based on \mathcal{D} .

Use optimization algorithms to optimize the acquisition function and obtain candidate solutions x^* .

Evaluate the objective function value of x^* and set $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x^i, f(x^i))\}_{i=1}^{N_i}$.

end

Return $\min_{x \in \mathcal{D}} f(x)$

Algorithm 1. GP surrogate models

Considering the limitations of traditional optimization algorithms in dealing with industrial system optimization problems, this paper proposes a new optimization algorithm called AG-MOPSO-GPS model, as shown in Figure 1. In contrast to the conventional GP surrogate model framework, we introduce

the SGP model and propose the SGP-assisted optimization method. The SGP surrogate model effectively addresses the computational overhead caused by the large data requirements of traditional gaussian process model. Moreover, we divide the optimization problem into multiple regions through grid partitioning and utilize the MOPSO algorithm to optimize particles in each region. By combining the advantages of grid partitioning and particle swarm optimization, we overcome the limitations of traditional optimization algorithms in dealing with multi-objective problems. At each iteration, the surrogate model is used to predict the next optimal solution and integrated into the subsequent iteration. The advantage of this algorithm lies in its ability to reduce computation costs by avoiding frequent evaluations of the true objective function. The AG-MOPSO-GPS algorithm is characterized by a broad search range and fast search speed due to the use of MOPSO. Additionally, it can effectively handle high-dimensional non-linear problems, and is supported by mathematical theory. The specific implementation of the algorithm includes key steps, such as selecting specific hyperparameters, sampling strategies, and stopping conditions, to improve accuracy and efficiency.

The steps of the AG-MOPSO-GPS model are as follows:

Step 1: Initialize population: randomly generate a certain number of particles and assign initial position and velocity values.

Step 2: Particle fitness evaluation: substitute the particles in the population into the objective function and calculate the fitness value of each particle.

Step 3: Update non-dominated solution set: determine whether the solution of the objective function is a non-dominated solution based on the dominance relation and update the non-dominated solution set.

Step 4: Grid division: according to the distribution of the non-dominated solution set, divide the grid and optimize each grid as a region.

Step 5: Update particle velocity and position based on non-dominated solution set: use the best particle position in each grid as a reference point, calculate the velocity and position of each particle based on the reference point, and update the position and velocity of the particles. At the same time, use the SGP surrogate method to optimize the reference point and predict the reference point for the next iteration.

Step 6: Record the best historical solution: record the best particle position and corresponding fitness value in each grid.

Step 7: Determine stopping criteria: determine whether the defined stopping criteria such as maximum number of iterations are met.

Step 8: Output results: output the final optimization results, including the position of particles in the population and their corresponding fitness values.

4 Results and discussions

In this section, we make a series of comparisons of surrogate model assisted multi-objective optimization algorithms to explore the advantages of surrogate model applied to complex optimization problems. Firstly, we compared the evaluation accuracy of surrogate

models on different functions. Secondly, we applied the surrogate models to multi-objective optimization algorithms and compared the performance of surrogate model-assisted optimization algorithms.

4.1 Testing of multi-objective optimization algorithm based on surrogate model

The Zitzler Deb Thiele (ZDT) test function set is a classic set of test functions for assessing the performance of multi-objective optimization algorithms (Lim et al., 2015). Developed by Zitzler and his colleagues at the University of Barcelona in 2000, the ZDT function set features independent objective functions with varying feasible range widths and non-linear characteristics. These properties make it suitable for evaluating the performance of PF search algorithms. The ZDT function set includes several functions, with ZDT1 and ZDT2 being the most widely used ones.

The problem definition of ZDT1 is as follows:

$$\begin{aligned}
 f_1(x) &= x_1 \\
 f_2(x) &= g(x) \left[1 - \sqrt{\frac{f_1(x)}{g(x)}} \right]
 \end{aligned}
 \tag{20}$$

Where, $g(x) = 1 + (9 \sum_{i=2}^n x_i) / (n - 1)$; $x = (x_1, \dots, x_n)^T \in [0, 1]^n$, which PF is convex.

The problem definition of ZDT2 is as follows:

$$\begin{aligned}
 f_1(x) &= x_1 \\
 f_2(x) &= g(x) \left[1 - \left(\frac{f_1(x)}{g(x)} \right)^2 \right]
 \end{aligned}
 \tag{21}$$

Where, ZDT2 takes values of $g(x)$ and x that both range and dimensionally are the same as ZDT1.

The problem definition of ZDT3 is as follows:

$$\begin{aligned}
 f_1(x) &= x_1 \\
 f_2(x) &= g(x) \left[1 - \sqrt{\frac{f_1(x)}{g(x)}} - \frac{f_1(x)}{g(x)} \sin(10\pi x_1) \right]
 \end{aligned}
 \tag{22}$$

Where, the PF of ZDT3 is discontinuous and has different scales on both targets.

DTLZ is another classic test function set used for testing multi-objective optimization algorithms (Li et al., 2015). Developed by Kalyanmoy Deb and others in 2002, this function set consists of a range of multi-objective test functions. Its features include high dimensions and interaction design in the objective functions, making it suitable for assessing the high-dimensional PF search capability.

The problem definition of DTLZ1 is as follows:

$$\begin{aligned}
 f_1(x) &= \frac{1}{2} \prod_{i=1}^{m-j} x_i (1 + g(x_{m:n})) \\
 f_{j=2:m-1}(x) &= \frac{1}{2} \left(\prod_{i=1}^{m-j} x_i \right) (1 - x_{m-j+1}) (1 + g(x_{m:n})) \\
 f_m(x) &= \frac{1}{2} (1 - x_1) (1 + g(x_{m:n}))
 \end{aligned}
 \tag{23}$$

Where, $g(x_{m:n}) = 100 [k + \sum_{i=n}^m ((x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)))]$, $x = (x_1, \dots, x_n)^T \in [0, 1]^n$, DTLZ1 has many local optimal solutions, and the PF of DTLZ1 satisfies $(\sum_{i=1}^m f_i(x)) = 0.5$.

The problem definition of DTLZ2 is as follows:

$$\begin{aligned}
 f_1(x) &= (1 + g(x_{m:n})) \prod_{i=1}^{m-1} \cos\left(\frac{x_i \pi}{2}\right) \\
 f_{j=2:m-1}(x) &= (1 + g(x_{m:n})) \prod_{i=1}^{m-1} \cos\left(\frac{x_i \pi}{2}\right) \sin\left(\frac{x_{m-j+1} \pi}{2}\right) \\
 f_m(x) &= (1 + g(x_{m:n})) \sin\left(\frac{x_1 \pi}{2}\right)
 \end{aligned}
 \tag{24}$$

Where, $g(x) = \sum_{i=n}^m ((x_i - 0.5)^2)$, The PF of DTLZ2 satisfies $(\sum_{i=1}^m f_i(x)^2) = 1$.

DTLZ7 problem has 2^{m-1} disconnected optimal Pareto regions, used to test the algorithm's ability to maintain subpopulations within different Pareto regions. The problem definition of DTLZ7 is as follows:

$$\begin{aligned}
 f_{j=1:m-1}(x) &= x_j \\
 f_m(x) &= (1 + g(x_{m:n})) \left(m - \sum_{i=1}^{m-1} \left[\frac{1}{1 + g(x_{m:n})} (1 + \sin(3\pi f_i(x))) \right] \right)
 \end{aligned}
 \tag{25}$$

Where, $g(x_{m:n}) = 1 + 9 \sum_{i=m-n}^n x_i / k$, $x = (x_1, \dots, x_n)^T \in [0, 1]^n$.

The WFG test functions encompass a total of 9 test problems, with each function exhibiting distinct characteristics and complexities to cover various types of multi-objective optimization problems (Li et al., 2018). In this paper, we have selected three test functions: WFG1, known for its plateau-like preference characteristics; WFG2, which demonstrates multi-modal non-continuity properties; and WFG3, a deceptive problem.

In general, ZDT test functions are suitable for evaluating relatively simple multi-objective optimization algorithms. Compared to ZDT, DTLZ test functions are more suitable for exploring algorithm performance under different problem characteristics, while WFG test functions offer more diversity and complexity, making them closer to real-world multi-objective optimization problems.

4.2 Evaluation criteria

The purpose of multi-objective optimization algorithm is to quickly search and approximate the true PF and distribute uniformly on it. Therefore, evaluating the performance of a multi-objective optimization algorithm requires consideration of its convergence, distribution, coverage, and search speed, among others. This paper tested the performance of the algorithm, mainly using two comprehensive indicators: inverted generational distance (IGD) and hypervolume (HV).

In multi-objective optimization, the IGD is an important indicator to evaluate the convergence, distribution uniformity, and generality of solutions. It is the inverse mapping of the generational distance (GD) and calculates the average distance from each individual in the true Pareto optimal solution set to the non-dominated solution set obtained by the algorithm. The

smaller the IGD value of the solution set, the better the performance. In addition to reflecting the convergence of solution sets, IGD can also reflect the distribution uniformity and generality of solution sets. A smaller IGD value indicates better diversity and convergence.

$$IGD(F^*, F) = \frac{\sum_{x \in F^*} \min dis(x, F)}{|F^*|} \quad (26)$$

Where, F is the non-dominated solution set obtained by the algorithm, F^* represents the true Pareto non-dominated solution set, $d(\cdot)$ represents the Euclidean distance, $|F^*|$ represents the number of solutions in the true PF.

The HV indicator is one of the metrics used to evaluate the comprehensive performance of multi-objective optimization algorithms. It represents the volume of the region in the objective space between the non-dominated solution set obtained by the algorithm and a reference point. HV can be used to measure the degree of closeness between the solution set and the optimal solution set and partially reflects the distribution of solutions in the objective space. A larger HV value indicates better comprehensive performance of the algorithm.

$$HV(P) = VOL\left(\bigcup_{x \in P} [x_1, z_1^r] \times \dots \times [x_m, z_m^r]\right) \quad (27)$$

Where, VOL represents the Lebesgue measure, m represents the number of non-dominated solutions, x_m represents the m th objective component of the solution set x in P , z_m^r represents the r th objective component of the reference point in the objective space.

4.3 Validation analysis of AG-MOPSO-GPS

Before validating the surrogate-assisted optimization algorithm, it is necessary to set the parameters of the optimization algorithm and the test functions. For the ZDT series test functions, the decision variables of ZDT1 and ZDT2 are set to 30. For the DTLZ series test functions, DTLZ1 has 7 decision variables and 3 dimensions, while DTLZ2 has 12 decision variables and 4 dimensions. For the WFG test function, the effectiveness of the method proposed in this article is verified by testing under three and four objectives. And for the three objectives problem, set 100 reference vectors, and for the four objectives problem, set 120 reference vectors with a vector dimension of 11 dimensions. The population size of the AG-MOPSO algorithm is set to 100, the number of partitions of each dimension in the adaptive grid-based method is set to 30, the inertia weight w is set to 0.4, the learning factor is set to 2.0, and the mutation parameter is set to 0.5. All the simulation experiments in this section were performed on a computer with 2.90 GHz AMD Ryzen 74800H with Radeon Graphics and 16 GB of memory using MATLAB_R2021b. The AG-MOPSO-GPS algorithm was run 30 times on uniformly random test functions to reduce accidental results, and the IGD measure and HV measure were calculated for each run. The data in Tables 1, 2 show the IGD and HV measures of the proposed algorithm and four other multi-objective algorithms, where “Mean” represents the average value of IGD and HV obtained by running the algorithm 30 times, and “Std” represents the standard deviation.

To better validate the superiority of the proposed algorithm in this paper, we selected two basic optimization algorithms, AG-MOPSO and NSGA-II, as well as two agent-based optimization algorithms, AG-MOPSO-ANNS and AG-MOPSO-SVMS, for comparative research. Tables 2, 3 present the statistical results of these five optimization algorithms on three sets of test functions concerning IGD and HV measures. The best average results are highlighted in bold. As revealed in Table 3, the proposed algorithm exhibits the best IGD results among different test functions, except for the ZDT3 test function. Particularly, in the ZDT1 and ZDT2 test functions, the IGD has a significant advantage of one order of magnitude. Although the IGD measure is not optimal for ZDT3, the proposed method still outperforms traditional optimization algorithms and SVM agent models, indicating that the AG-MOPSO-GPS algorithm can obtain an optimal solution set closer to the true PF, further validating its effectiveness. Furthermore, the proposed algorithm shows significant improvement in the DTLZ type of test functions compared to other optimization algorithms, demonstrating its generalization and robustness for three-objective high-complexity optimization problems. Additionally, in WFG test functions with varying numbers of objectives, the proposed algorithm achieves the smallest IGD values, indicating that the adaptive grid strategy and SGP agent model of AG-MOPSO-GPS can achieve more stable optimization results while ensuring convergence and diversity, making it suitable for solving high-dimensional multi-objective optimization problems. Interestingly, regardless of whether the ANN or SVM agent model is used in different test functions, the IGD measure is superior to AG-MOPSO and NSGA-II, indicating the effectiveness of the agent-assisted optimization method in solving multi-objective optimization problems.

Meanwhile, the average and standard deviation of the approximate PF obtained using different optimization algorithms for ZDT, DTLZ and WFG test functions based on the HV measure are shown in Table 3. The HV measure can reflect the comprehensive performance in terms of convergence and diversity. It can be observed that the average values obtained by the AG-MOPSO-GPS algorithm are better than those of the other four algorithms, while the standard deviation is better than that of the other algorithms in most cases. This indicates that the proposed algorithm can obtain an optimal solution set that is closer to the true PF in various forms of multi-objective optimization problems, effectively avoiding the algorithm from getting stuck in local optima and demonstrating good global search and local development performance.

To more intuitively compare the convergence and distribution of different algorithms, Figures 2–4 present the PF obtained by different optimization algorithms on different test functions. It can be clearly seen that for the ZDT test function, the non-dominated solutions generated by the AG-MOPSO and NSGA-II algorithms are still far from the true PF of ZDT. However, the agent-assisted multi-objective optimization algorithm can converge to the true PF of the ZDT test function. As for DTLZ1, it has multiple local PFs, and the agent-assisted multi-objective optimization algorithm can escape from local optima, which successfully finds solutions on the true PF. Comparing the ANN and SVM agent models with GP agent model-assisted multi-objective optimization algorithms, it can be

found that the GP agent-assisted algorithm can converge better to the true PF. In addition, the PF obtained by the proposed algorithm on the DTLZ2 test function is well-distributed in the objective space, and there are more solutions, which is better than the other compared algorithms. Similarly, for the DTLZ7 test function with a discontinuous PF, the proposed method also achieves a well-distributed and abundant set of non-dominated solutions. In the WFG test functions, the proposed algorithm not only attains a smaller IGD value but also demonstrates better convergence and diversity in the obtained solution set. Overall, the other algorithms' non-dominated solution sets still fail to fit the true PF effectively, requiring more iterations and incurring higher computational costs.

5 Engineering cases

With the increasingly serious global climate problems, the development of renewable energy sources is urgent, among which wind power is the main force of clean energy. Therefore, improving the capacity of wind farms is a pressing issue. Generally, for wind farms that have been built, improving the power generation efficiency can be achieved by improving the control strategy of wind turbines, while for planned wind farms, the power generation capacity can be improved by optimizing the layout of wind turbines. Wind farm layout optimization is a highly complex multi-objective optimization problem, involving multiple fields. This paper applies the agent-based optimization framework to wind farm layout optimization and proves its superiority in complex optimization problems by comparing with different optimization algorithms.

The wind farm layout optimization problem aims to obtain a wind turbine layout that minimizes the annual energy cost f_1 , minimizing the overall wind farm area f_2 , reducing wake loss and maximizing efficiency f_3 through a multi-objective strategy. The specific expressions are as follows:

$$\begin{aligned}
 F(\mathbf{x}, \mathbf{y}) &= \min(f_1, f_2, -f_3) \\
 f_1 &= \frac{N_t \left(\frac{2}{3} + \frac{1}{3} e^{-0.00174N_t^2} \right)}{8760 \sum_{k=0}^{360} \sum_{i=1}^{N_t} f_k(\theta) P_i(U_i(\theta))} \\
 f_2 &= \frac{1}{2} abs \left(\sum_{i=1}^{n-1} x_i y_{i+1} + x_n y_1 - \sum_{i=1}^{n-1} x_{i+1} y_i + x_1 y_n \right) \\
 f_3 &= 100 \frac{\sum_{k=0}^{360} \sum_{i=1}^{N_t} f_k(\theta) P_i(U_i(\theta))}{\sum_{k=0}^{360} \sum_{i=1}^{N_t} f_k(\theta) P_i(U_0(\theta))}
 \end{aligned} \tag{28}$$

Subject to

$$\begin{aligned}
 \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} \max \left(\left(6D_{wt} - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \right), 0 \right) &\leq 0 \\
 x_i &\geq D_{wt}; x_i \leq 2000 - D_{wt} \\
 y_i &\geq D_{wt}; y_i \leq 2000 - D_{wt} \\
 x_i &\neq x_j; y_i \neq y_j
 \end{aligned} \tag{29}$$

Where, N_t is the total number of wind turbines; $f_k(\theta)$ represents the wind speed probability in the θ direction, with a division step of 10° . $U_i(\theta)$ is the wind speed in the θ direction. $P_i(U_i(\theta))$ represents the power output under $U_i(\theta)$. $P_i(U_0(\theta))$ represents the power output obtained in the free stream wind speed (excluding wake loss). D_{wt} is the diameter of the wind turbine. The specific details of wind farm parameters and experimental settings can be found in ref. (Moreno et al., 2021).

The proposed algorithm in this paper is applied to case 1 in Moreno et al. (2021), and compared with the best-performing algorithm MO-LSA under the fixed wind speed (12 m/s) condition, and the best-performing algorithm NSGA-II under the variable wind speed (8, 12, 17 m/s) condition. The performance is shown in Table 4, and it can be observed that the AG-MOPSO-GPS algorithm proposed in this paper is superior in terms of IGD and HV measures in all cases.

Figures 5, 6 are optimization comparison diagrams between the optimal method in the literature and the method proposed in this paper. It can be found that the method proposed in this paper obtains more non-dominated solutions and a more uniform distribution when optimizing the layout of wind farms. The optimal solutions of each optimization algorithm were substituted into the objective function to obtain the annual energy cost, wind farm area, and wind turbine efficiency, as shown in Table 5. Compared with the method proposed in this paper, MO-LSA reduced the energy cost by 5.48%, decreased the wind farm area by 10.12%, and improved the wind turbine efficiency by 3.02% under the constant wind speed condition. NSGA-II algorithm reduced the energy cost by 2.71%, decreased the wind farm area by 3.93%, and improved the wind turbine efficiency by 1.60% compared with the method proposed in this paper under the variable wind speed condition. This indicates that developing high-performance optimization algorithms is important for improving efficiency and reducing costs in multi-objective optimization problems in practical industrial applications.

6 Conclusion and future plans

With the continuous improvement of industrial demand, the complexities arising from the interrelatedness of different optimization objectives are increasing. Therefore, the development of efficient methods to solve complex optimization problems is of paramount importance in the realm of multi-objective optimization. In this paper, we thoroughly analyze and study the problems of the multi-objective optimization problems requiring a large number of function evaluations and propose a new method to solve complex optimization problems by combining the AG-MOPSO algorithm with surrogate models. Through comparative research, the main findings are as follows.

1. Validation on different types of test functions demonstrates that the proposed AG-MOPSO-GPS achieves satisfactory results in various metrics. This indicates that the proposed method in this paper can obtain a more realistic PF. Specifically, in the WFG test functions, the ICD and HV

metrics outperform other algorithms significantly. This suggests that when multi-objective optimization problems have multiple local optima, the proposed method in this paper effectively assists particles in capturing and maintaining these multiple Pareto optimal solution sets.

2. Compared with the traditional optimization models AG-MOPSO and NSGA-II, the surrogate model assisted multi-objective optimization algorithm can obtain non-dominated solutions and the distribution of solutions is more uniform, which indicates that the surrogate model can provide a more efficient, comprehensive and robust solution for multi-objective optimization problems, which helps to accelerate the optimization process and obtain a better approximate PF.
3. From the PF and solutions distribution, the SGP surrogate model can provide uncertainty estimates of unknown target function values, helping the optimization algorithm be more robust when dealing with noise or uncertainty.
4. In the wind farm layout optimization, this method can effectively reduce the land use area, reduce costs, and improve power generation efficiency. This indicates that for complex problems with multiple conflicting objectives, the proposed method in this paper can efficiently search for the Pareto optimal solution set, making it suitable for tackling complex multi-objective optimization problems.

In generally, the proposed approach in this paper can quickly solve multi-objective optimization problems and is suitable for various industrial requirements, demonstrating strong robustness. Future research plans include exploring online updating techniques for agent models to adapt to dynamic characteristics of optimization problems that change over time or parameters and to timely reflect the regularity of objective functions, in order to improve the real-time and accuracy of agent models.

References

- Akinola, O. O., Ezugwu, A. E., Agushaka, J. O., Zitar, R. A., and Abualigah, L. (2022). Multiclass feature selection with metaheuristic optimization algorithms: a review. *Neural Comput. Appl.* 34 (22), 19751–19790. doi:10.1007/s00521-022-07705-4
- Avendaño-Valencia, L. D., Abdallah, I., and Chatzi, E. (2021). Virtual fatigue diagnostics of wake-affected wind turbine via Gaussian Process Regression. *Renew. Energy* 170, 539–561. doi:10.1016/j.renene.2021.02.003
- Chen, G., Zhang, K., Xue, X., Zhang, L., Yao, C., Wang, J., et al. (2022). A radial basis function surrogate model assisted evolutionary algorithm for high-dimensional expensive optimization problems. *Appl. Soft Comput.* 116, 108353. doi:10.1016/j.asoc.2021.108353
- Cicczazzo, A., Di Pillo, G., and Latorre, V. (2015). A SVM surrogate model-based method for parametric yield optimization. *IEEE Trans. Computer-Aided Des. Integr. Circuits Syst.* 35 (7), 1224–1228. doi:10.1109/tcad.2015.2501307
- Coello, C. A. C., Pulido, G. T., and Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Trans. Evol. Comput.* 8 (3), 256–279. doi:10.1109/tevc.2004.826067
- Cui, Y., Geng, Z., Zhu, Q., and Han, Y. (2017). Review: multi-objective optimization methods and application in energy saving. *Energy* 125, 681–704. doi:10.1016/j.energy.2017.02.174
- Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *J. Am. Stat. Assoc.* 86 (416), 953–963. doi:10.1080/01621459.1991.10475138
- Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature* 521 (7553), 452–459. doi:10.1038/nature14541
- Giovanis, D. G., and Shields, M. D. (2020). Data-driven surrogates for high dimensional models using Gaussian process regression on the Grassmann manifold. *Comput. Methods Appl. Mech. Eng.* 370, 113269. doi:10.1016/j.cma.2020.113269
- Golparvar, B., Papadopoulos, P., Ezzat, A. A., and Wang, R.-Q. (2021). A surrogate-model-based approach for estimating the first and second-order moments of offshore wind power. *Appl. Energy* 299, 117286. doi:10.1016/j.apenergy.2021.117286
- Grimstad, B., Foss, B., Heddle, R., and Woodman, M. (2016). Global optimization of multiphase flow networks using spline surrogate models. *Comput. Chem. Eng.* 84, 237–254. doi:10.1016/j.compchemeng.2015.08.022
- Gu, Q., Wang, Q., Li, X., and Li, X. (2021). A surrogate-assisted multi-objective particle swarm optimization of expensive constrained combinatorial optimization problems. *Knowledge-Based Syst.* 223, 107049. doi:10.1016/j.knsys.2021.107049
- Han, Z.-H., and Zhang, K.-S. (2012). Surrogate-based optimization. *Real-world Appl. Genet. Algorithms* 343. doi:10.48550/arXiv.2105.03893
- He, C., Zhang, Y., Gong, D., and Ji, X. (2023). A review of surrogate-assisted evolutionary algorithms for expensive optimization problems. *Expert Syst. Appl.* 217, 119495. doi:10.1016/j.eswa.2022.119495
- Jeong, S., Murayama, M., and Yamamoto, K. (2005). Efficient optimization design method using kriging model. *J. Aircr.* 42 (2), 1375–1420. doi:10.2514/1.c10485e

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

Author contributions

YC: Methodology, software, writing-original draft, visualization
 LW: Methodology, Validation, writing-review and editing
 HH: Validation, writing-review and editing. All authors contributed to the article and approved the submitted version.

Funding

This work is supported by the Hunan Provincial Education Planning Project (XJK23CGD053).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *J. Glob. Optim.* 13, 455–492. doi:10.1023/a:1008306431147
- Joseph, V. R., Hung, Y., and Sudjianto, A. (2008). *Blind kriging: A new method for developing metamodels*. New York City, United States: ASME.
- Kudela, J., and Matousek, R. (2022). Recent advances and applications of surrogate models for finite element method computations: A review. *Soft Comput.* 26 (24), 13709–13733. doi:10.1007/s00500-022-07362-8
- Li, K., Wang, R., Zhang, T., and Ishibuchi, H. (2018). Evolutionary many-objective optimization: A comparative study of the state-of-the-art. *IEEE Access* 6, 26194–26214. doi:10.1109/access.2018.2832181
- Li, Y., Liu, H., Xie, K., and Yu, X. (2015). “A method for distributing reference points uniformly along the Pareto front of DTLZ test functions in many-objective evolutionary optimization,” in 2015 5th International Conference on Information Science and Technology (ICIST), Changsha, China, 24–26 April 2015, 541–546.
- Lim, W. J., Jambek, A. B., and Neoh, S. C. (2015). Kursawe and ZDT functions optimization using hybrid micro genetic algorithm (HMGA). *Soft Comput.* 19, 3571–3580. doi:10.1007/s00500-015-1767-5
- Ling, C., Kuo, W., and Xie, M. (2022). Complementary and alternative medicine during COVID-19 pandemic: what we have done. *IEEE Trans. Reliab.* 20, 1–3. doi:10.1016/j.joim.2021.11.008
- Liu, B., Koziel, S., and Zhang, Q. (2016). A multi-fidelity surrogate-model-assisted evolutionary algorithm for computationally expensive optimization problems. *J. Comput. Sci.* 12, 28–37. doi:10.1016/j.jocs.2015.11.004
- Liu, B., Zhang, Q., and Gielen, G. G. (2013). A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Trans. Evol. Comput.* 18 (2), 180–192. doi:10.1109/tevc.2013.2248012
- Liu, F., Zhang, Q., and Han, Z. (2023b). MOEA/D with gradient-enhanced kriging for expensive multiobjective optimization. *Nat. Comput.* 22 (2), 329–339. doi:10.1007/s11047-022-09907-0
- Liu, J., Wang, Y., Sun, G., and Pang, T. (2023a). Solving highly expensive optimization problems via evolutionary expected improvement. *IEEE Trans. Syst. Man, Cybern. Syst.* 53, 4843–4855. doi:10.1109/tsmc.2023.3257030
- Liu, J.-Q., Feng, Y.-W., Xue, X.-F., and Lu, C. (2021). Intelligent extremum surrogate modeling framework for dynamic probabilistic analysis of complex mechanism. *Math. Problems Eng.* 2021, 1–12. doi:10.1155/2021/6681489
- Liu, X., Zhao, W., and Wan, D. (2022). Multi-fidelity Co-Kriging surrogate model for ship hull form optimization. *Ocean Eng.* 243, 110239. doi:10.1016/j.oceaneng.2021.110239
- Lystad, T. M., Fenerci, A., and Øiseth, O. (2023). Full long-term extreme buffeting response calculations using sequential Gaussian process surrogate modeling. *Eng. Struct.* 292, 116495. doi:10.1016/j.engstruct.2023.116495
- Mavrouniotis, M., Li, C., and Yang, S. (2017). A survey of swarm intelligence for dynamic optimization: algorithms and applications. *Swarm Evol. Comput.* 33, 1–17. doi:10.1016/j.swevo.2016.12.005
- Moreno, S. R., Pierzezan, J., dos Santos Coelho, L., and Mariani, V. C. (2021). Multi-objective lightning search algorithm applied to wind farm layout optimization. *Energy* 216, 119214. doi:10.1016/j.energy.2020.119214
- Nguyen, A.-T., Reiter, S., and Rigo, P. (2014). A review on simulation-based optimization methods applied to building performance analysis. *Appl. energy* 113, 1043–1058. doi:10.1016/j.apenergy.2013.08.061
- Palmer, S. (2019). *Evolutionary algorithms and computational methods for derivatives pricing*. Ucl. London: University College London.
- Parsopoulos, K. E., and Vrahatis, M. N. (2008). *Multi-objective optimization in computational intelligence: Theory and practice*. Pennsylvania, United States: IGI global, 20–42.
- Preen, R. J., Bull, L., and Adamatzky, A. (2019). Towards an evolvable cancer treatment simulator. *Biosystems* 182, 1–7. doi:10.1016/j.biosystems.2019.05.005
- Satria Palar, P., Rizki Zuhail, L., and Shimoyama, K. (2020). Gaussian process surrogate model with composite kernel learning for engineering design. *AIAA J.* 58 (4), 1864–1880. doi:10.2514/1.j058807
- Shadab, S., Hozefa, J., Sonam, K., Wagh, S., and Singh, N. M. (2022). Gaussian process surrogate model for an effective life assessment of transformer considering model and measurement uncertainties. *Int. J. Electr. Power and Energy Syst.* 134, 107401. doi:10.1016/j.ijepes.2021.107401
- Su, G., Peng, L., and Hu, L. (2017). A Gaussian process-based dynamic surrogate model for complex engineering structural reliability analysis. *Struct. Saf.* 68, 97–109. doi:10.1016/j.strusafe.2017.06.003
- Sun, C., Jin, Y., Cheng, R., Ding, J., and Zeng, J. (2017). Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Trans. Evol. Comput.* 21 (4), 644–660. doi:10.1109/tevc.2017.2675628
- Talgorn, B., Audet, C., Le Digabel, S., and Kokkolaras, M. (2018). Locally weighted regression models for surrogate-assisted design optimization. *Optim. Eng.* 19, 213–238. doi:10.1007/s11081-017-9370-5
- Vafadar, A., Guzzomi, F., Rassau, A., and Hayward, K. (2021). Advances in metal additive manufacturing: A review of common processes, industrial applications, and current challenges. *Appl. Sci.* 11 (3), 1213. doi:10.3390/app11031213
- Wang, C., Qiang, X., Xu, M., and Wu, T. (2022b). Recent advances in surrogate modeling methods for uncertainty quantification and propagation. *Symmetry* 14 (6), 1219. doi:10.3390/sym14061219
- Wang, Z., Sun, S., and Ding, Y. (2022a). Fatigue optimization of structural parameters for orthotropic steel bridge decks using RSM and NSGA-II. *Math. Problems Eng.* 2022, 1–12. doi:10.1155/2022/4179898
- Yang, J.-J., Zhou, J.-Z., Fang, R.-C., Li, Y.-H., and Liu, L. (2008). Multi-objective particle swarm optimization based on adaptive grid algorithms. *J. Syst. Simul.* 20 (21), 5843–5847.
- Zhang, Z., Ye, L., Qin, H., Liu, Y., Wang, C., Yu, X., et al. (2019). Wind speed prediction method using shared weight long short-term memory network and Gaussian process regression. *Appl. energy* 247, 270–284. doi:10.1016/j.apenergy.2019.04.047
- Zhao, H., Gao, Z.-H., and Xia, L. (2022). Efficient aerodynamic analysis and optimization under uncertainty using multi-fidelity polynomial chaos-Kriging surrogate model. *Comput. Fluids* 246, 105643. doi:10.1016/j.compfluid.2022.105643
- Zheng, N., Wang, H., and Yuan, B. (2022). An adaptive model switch-based surrogate-assisted evolutionary algorithm for noisy expensive multi-objective optimization. *Complex and Intelligent Syst.* 8 (5), 4339–4356. doi:10.1007/s40747-022-00717-6
- Zhou, Z., Ong, Y. S., Nair, P. B., Keane, A. J., and Lum, K. Y. (2006). Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Trans. Syst. Man, Cybern. Part C Appl. Rev.* 37 (1), 66–76. doi:10.1109/tsmcc.2005.855506