An Uncertainty Measure for Prediction of Non-Gaussian Process Surrogates

Caie Hu caie_hu@cug.edu.cn
School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan 430074, China

Sanyou Zeng* sanyouzeng@gmail.com
School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan 430074, China

Changhe Li* changhe.lw@gmail.com
School of Automation, China University of Geosciences, Wuhan 430074, China
Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan, 430074, China
Engineering Research Center of Intelligent Technology for Geo-Exploration, Ministry of Education, Wuhan, 430074, China

https://doi.org/10.1162/evco_a_00316

Abstract

Model management is an essential component in data-driven surrogate-assisted evolutionary optimization. In model management, the solutions with a large degree of uncertainty in approximation play an important role. They can strengthen the exploration ability of algorithms and improve the accuracy of surrogates. However, there is no theoretical method to measure the uncertainty of prediction of Non-Gaussian process surrogates. To address this issue, this article proposes a method to measure the uncertainty. In this method, a stationary random field with a known zero mean is used to measure the uncertainty of prediction of Non-Gaussian process surrogates. Based on experimental analyses, this method is able to measure the uncertainty of prediction of Non-Gaussian process surrogates. The method's effectiveness is demonstrated on a set of benchmark problems in single surrogate and ensemble surrogates cases.

Keywords

Evolutionary computation, data-driven evolutionary optimization, surrogate, model management, Non-Gaussian process.

1 Introduction

Data-driven optimization problems usually involve objective and constraint functions that are not available, and the evaluation of these functions is time-consuming and complex. There are only small data from physical experiments, numerical simulations, or daily life, and the evaluation of these functions involves a number of computationally expensive numerical simulations or costly physical experiments (Preen and Bull, 2016; Wang et al., 2016; Jin et al., 2018).

*Corresponding author.

Manuscript received: 19 April 2021; revised: 22 April 2021, 10 January 2022, 23 August 2022; accepted: 16 September 2022.

Evolutionary algorithms (EAs) are population-based search methods that mimic natural biological evolution and species' social behavior. They are promising in solving non-convex, constrained, multiobjective, or dynamic problems (Michalewicz and Schoenauer, 1996; Hart et al., 1998; Li et al., 2014; Zhang, Mei et al., 2021). However, most existing research on EAs usually assumes that the analytic objective and constraint functions are available, and evaluating these functions is cheap and simple. Therefore, EAs cannot be directly used to solve the data-driven optimization problems. Surrogate-assisted evolutionary algorithms (SAEAs) are considered to address the limitation of EAs in solving these problems (Jin et al., 2000; Tong et al., 2019; Zhang, Li et al., 2021; Wang et al., 2022). In SAEAs, many machine learning models can be used as surrogates to approximate the exact functions, including polynomial regression (PR), Gaussian process (GP), artificial neural network (ANN), radial basis function network (RBFN), support vector machine (SVM), and the ensemble of these surrogates. A limited number of exact function evaluations are carried out, and a small amount of data is used to train these surrogates (Braun et al., 2009; Jin et al., 2000; Chugh et al., 2019).

For all surrogates mentioned above, GP is usually used (Emmerich et al., 2006; Coelho and Bouillard, 2011; Chugh et al., 2016; Zhan and Xing, 2021). There is provided prediction and uncertainty information by GP, which is important in SAEAs. Then the existing infill sampling criteria can be used to guide the search of EAs and the update of surrogates, such as the lower confidence bound (LCB) (Torczon and Trosset, 1998), the expected improvement (EI) (Jones et al., 1998) and the probability of improvement (PoI) (Ulmer et al., 2003). On the contrary, although many Non-Gaussian process (Non-GP) surrogates can also provide a good prediction, they cannot provide the uncertainty of prediction of surrogates. In this case, these Non-GP surrogates have significant limitations: (1) Because there is no uncertainty information of prediction of surrogates, it is hard to improve the exploration of EAs and the accuracy of surrogates; (2) The existing infill sampling criteria cannot be used to guide the search of EAs.

It should be emphasized that the uncertainty information of prediction of surrogates plays an essential role in model management in SAEAs, because (1) solutions with a large degree of uncertainty indicate that the fitness landscape around them has not been well explored, and therefore the evaluation of these solutions is likely to find a better solution (Branke and Schmidt, 2005); (2) evaluating these solutions can most effectively improve the accuracy of surrogates (Jin, 2011).

Several methods are used to measure the uncertainty of prediction of Non-GP surrogates. For instance, Bayesian neural networks can measure the uncertainty of prediction of neural networks (Gal and Ghahramani, 2015). Cross-validation also can be used to measure the uncertainty of prediction of surrogates (Hutter et al., 2019). However, there is a significant limitation for the two methods: the accuracy of uncertainty highly depends on the size of training data. However, there is not much training data in data-driven optimization progress. Besides, the prior distribution also needs to be known for Bayesian Neural Networks. Based on the limitation, the two methods will not be investigated in the article.

In addition to the above methods, there are also three typical methods: (1) The distance from the solutions to the existing training data has been used as an uncertainty measure in Branke and Schmidt (2005). Since ensemble surrogates have been proven to provide uncertainty information, two methods have been proposed to measure the uncertainty of prediction of ensemble surrogates. (2) The literature (Wang et al., 2017) defined the uncertainty measurement to be the maximum difference between outputs

of ensemble members. (3) The variance of predictions output by the base surrogates of ensemble is used to estimate the uncertainty of prediction of ensemble surrogates (Guo et al., 2018).

Among the three methods above, the first method is a qualitative uncertainty measurement method. In theory, it is not able to accurately measure the uncertainty of prediction of surrogates. Instead, it indicates only the crowded degree of the neighborhood of a solution. The second method is the disagreement among the outputs of ensemble members for the prediction of surrogates. This method was proposed based on Query-by-Committee (QBC) in active learning, which shows that the query with the maximum disagreement strategy can efficiently enhance the accuracy of surrogates (Wang et al., 2017). In essence, this method describes the difference of predictions among ensemble members in a solution. In the third method, the uncertainty of prediction of surrogates is defined by the variance of predictions output by the base surrogates of ensemble. It indicates the average squared deviation of the base surrogates about the output of ensemble. In the probability and statistic viewpoint, these methods for measuring the uncertainty of prediction of Non-GP surrogates are not a sound method. These methods mentioned above cannot address one important issue: to measure the uncertainty of prediction of Non-GP surrogates. Therefore, it can be confirmed that there is no a theoretical sound method to measure the uncertainty of prediction of Non-GP surrogates.

To address the issue mentioned above, this article proposes an uncertainty measure for the prediction (UMP) of Non-GP surrogates. This method can be written in the form of a random field model. In detail, it consists of two components: regression function (namely Non-GP surrogate) and residual variation (also known as uncertainty). In this method, two components are uncorrelated. In the first term, Non-GP surrogate as regression function only depends on decision variables, and the second term represents the uncertainty of prediction of Non-GP surrogate based on a stationary random field. Thus, based on the random field model, the uncertainty of prediction of Non-GP surrogate can be measured. Then, the existing infill sampling criteria can be used to guide the search of algorithms and the update of surrogates.

In this article, an uncertainty measure for the prediction of Non-GP Surrogates is proposed to overcome the drawbacks of existing uncertainty methods. The main contribution of this article can be summarized as follows:

(1) An uncertainty measure for the prediction of Non-GP surrogates is proposed, which overcomes the drawbacks of existing uncertainty methods;

(2) The effectiveness of the proposed method is investigated on a set of benchmark problems and analysed on Rastrigin function in both single surrogate and ensemble surrogates cases. The experimental results demonstrate that the proposed method is promising in solving data-driven optimization problems.

The rest of this article is structured as follows. Section 2 presents a brief review of the used surrogates, ensemble surrogates, and infill sampling criterion in this article. Section 3 presents the proposed uncertainty measure for the prediction of Non-GP surrogates. Section 4 demonstrates and discusses experimental results. Finally, Section 5 concludes the article with a summary and looks into the future work.

2 Related Work

Surrogates and infill sampling criteria are essential components in online surrogate-assisted evolutionary algorithms. This section presents a brief review of surrogates and the infill sampling criterion involved in this article.

2.1 Polynomial Regression

Polynomial regression adopts the statistical tools of regression and analysis of variance to obtain the minimum variance of regression. It is widely used in approximating exact objective and constraint functions. The formulation of the polynomial regression at any untested $\boldsymbol{x}$ is defined as follows

$$\hat{f}(\boldsymbol{x}) = \beta_0 + \Sigma_{i=1}^{d}\beta_i x_i + \Sigma_{i=1,j=1,i\leq j}^{d}\beta_{i,j}x_i x_j + \Sigma_{i=1,j=1,k=1,i\leq j\leq k}^{d}\beta_{i,j,k}x_i x_j x_k + \cdots \quad (1)$$

where β_0, β_i, $\beta_{i,j}$, $\beta_{i,j,k}$ are the coefficients to be estimated, d is the dimension of problems; usually, the least square method (LSM) is often used to estimate these coefficients in the surrogate.

2.2 Radial Basis Function Network

Like other neural networks, RBFN has an input layer, hidden layer, and output layer. It uses radial basis functions as its activation functions. In RBFN, the input layer is directly connected to the hidden one, and the output of RBFN at an untested $\boldsymbol{x}$ has the following expression

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{M}\omega_i\psi(\parallel \boldsymbol{x} - \boldsymbol{c}_i \parallel_p), \quad (2)$$

where ψ is activation function; $\boldsymbol{c}_i$ can be any point vector (e.g., origin or center); M is the number of nodes of the hidden layer; and ω is the unknown weights to be estimated, which can be determined by LSM or backpropagation based on gradient descent; p is norm.

2.3 Support Vector Machine

SVM is one of the popular surrogates based on statistical learning theory and is often used as a surrogate by constructing a hyperplane in high-dimensional space. The SVM at an untested $\boldsymbol{x}$ is expressed as

$$\hat{f}(\boldsymbol{x}) = \boldsymbol{\omega}^T\boldsymbol{\phi}(\boldsymbol{x}) + b, \quad (3)$$

where $\boldsymbol{\phi}(\boldsymbol{x})$ is feature vector; coefficient vector $\boldsymbol{\omega}$ and coefficient b need to be estimated.

The unknown parameter $\boldsymbol{\omega}$ and b can be obtained by optimizing a constrained optimization problem (Cristianini and Shawe-Taylor, 2000) based on observed values y_i at $\boldsymbol{x}_i$ for $i = 1, \cdots, N$, which is shown as

$$\begin{aligned} min \quad & \frac{1}{2}\|\boldsymbol{\omega}\|^2 + L\sum_{i=1}^{N}(\xi_i + \xi_i') \\ st \quad & \begin{cases} y_i - \boldsymbol{\omega}^T\boldsymbol{\phi}(\boldsymbol{x}_i) - b \leq \varepsilon + \xi_i \\ \boldsymbol{\omega}^T\boldsymbol{\phi}(\boldsymbol{x}_i) + b - y_i \leq \varepsilon + \xi_i' \\ \xi_i', \xi_i \geq 0, \end{cases} \end{aligned} \quad (4)$$

where $L = 1.0$ and $\varepsilon = 0.1$ are prespecified values in this article, and ξ_i and ξ_i' are slack variables representing upper and lower constraints.

2.4 Ensemble Surrogates

The ensemble surrogates have been proven to outperform most of the single surrogates. They are able to generate more reliable predictions of fitness landscape of problems than single surrogates (Liu et al., 2000; Queipo et al., 2005), when little is known about

the problem to be optimized at hand. The prediction of ensemble surrogates $\hat{f}_{ens}(\boldsymbol{x})$ is formulated as

$$\hat{f}_{ens}(\boldsymbol{x}) = \sum_{i=1}^{K} w_i \hat{f}_i(\boldsymbol{x}), \sum_{i=1}^{K} w_i = 1, \tag{5}$$

where $\hat{f}_i(\boldsymbol{x})$ represents the output of the ith member in the ensemble; K is the number of members in the ensemble, $K = 3$ in this article; in this article, w_i is the weight of the ith member defined by

$$w_i = 0.5 - \frac{e_i}{2(\sum_{j=1}^{K} e_j)}, \tag{6}$$

where e_i and e_j are the root mean square error (RMSE) of the ith and jth member in the ensemble, respectively.

The ensemble surrogates also have been proven to provide uncertainty information of prediction of ensemble surrogates, and two methods have been proposed to measure the uncertainty information. The literature (Wang et al., 2017) defined the uncertainty measurement to be the maximum difference between outputs of ensemble members, as shown in Eq. (7).

$$U(\boldsymbol{x}) = \max(\hat{f}_i(\boldsymbol{x}) - \hat{f}_j(\boldsymbol{x})), \tag{7}$$

where the uncertainty $U(\boldsymbol{x})$ at $\boldsymbol{x}$ is the maximum difference between the outputs of two ensemble members $\hat{f}_i(\boldsymbol{x})$ and $\hat{f}_j(\boldsymbol{x})$.

The literature (Guo et al., 2018) used the variance of predictions output by the base members of ensemble to estimate the uncertainty of prediction of ensemble surrogates, as shown in Eq. (8).

$$U(\boldsymbol{x}) = \frac{1}{K-1} \sum_{i=1}^{K} (\hat{f}_i(\boldsymbol{x}) - \hat{f}_{ens}(\boldsymbol{x}))^2. \tag{8}$$

2.5 Lower Confidence Bound

The LCB was suggested (Lewis et al., 2000; Emmerich et al., 2002) to select potential candidate solutions, especially in solving multimodal optimization problems. LCB can prevent premature convergence and enhance the search toward less explored regions in search space. The expression of LCB is

$$f_{LCB}(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}) - \omega \hat{s}(\boldsymbol{x}), \tag{9}$$

where $\hat{f}(\boldsymbol{x})$ and $\hat{s}(\boldsymbol{x})$ are prediction mean and variance (uncertainty degree) from surrogates, respectively; the parameter ω scales the impact of the variance; a reasonable choice is $\omega = 2$, which leads to a high confidence probability (around 97%) (Emmerich et al., 2002).

3 Uncertainty Measure for Prediction of Non-GP Surrogates

We aim to address the issue that there is no theoretical method to measure the uncertainty of prediction of Non-GP surrogates. Hence, an uncertainty measure for prediction of Non-GP surrogates is proposed in this article. This method can be written in the form of random field model. In detail, it consists of two components: regression function (namely Non-GP surrogate) and residual variation (the uncertainty of prediction of Non-GP surrogate). In this method, the two components are uncorrelated. In the first term, Non-GP surrogate as regression function only depends on decision variables, and the second term represents the uncertainty of prediction of Non-GP surrogate based on a stationary random field.

3.1 Formulation for Uncertainty Measure for Prediction

The UMP is formulated as

$$F(\boldsymbol{x}) = m(\boldsymbol{x}) + \epsilon(\boldsymbol{x}), \tag{10}$$

where $m(\boldsymbol{x})$ as a regression function can be any Non-GP surrogate or ensemble Non-GP surrogates, $\epsilon(\boldsymbol{x})$ is a mean 0 random field with distribution $\mathcal{N}(0, \sigma^2)$.

In this article, the UMP makes the assumptions in building a cheap surrogate for an expensive function $y = f(\boldsymbol{x}), \boldsymbol{x} \in R^d$, $F(\boldsymbol{x}) \sim \mathcal{N}(m(\boldsymbol{x}), \sigma^2)$ is a random variable, and $\epsilon(\boldsymbol{x}) \sim \mathcal{N}(0, \sigma^2)$. For any $\boldsymbol{x}, \boldsymbol{x}' \in R^d$, the correlation between $\epsilon(\boldsymbol{x})$ and $\epsilon(\boldsymbol{x}')$, depends on the distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$. The correlation function $c(\boldsymbol{x}, \boldsymbol{x}')$ in this article is shown in Eq. (11).

$$c(\boldsymbol{x}, \boldsymbol{x}' \mid \boldsymbol{\theta}) = exp\left(-\sum_{i=1}^{d} \theta_i |x_i - x_i'|^2\right), \tag{11}$$

where d is dimension of problems; $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_d]^T$ measures the importance or activity of the variable $\boldsymbol{x}$.

3.1.1 Hyperparameter Estimation

In UMP, the hyperparameters σ^2, and $\boldsymbol{\theta}$ can be determined by maximizing the log likelihood function based on observe values y_i at $\boldsymbol{x}_i (i = 1, \cdots, N)$, which is shown as

$$-\frac{1}{2}[Nlog(2\pi\sigma^2) + log(det(\boldsymbol{C})) + (\boldsymbol{y} - \boldsymbol{m})^T \boldsymbol{C}^{-1}(\boldsymbol{y} - \boldsymbol{m})/\sigma^2], \tag{12}$$

where $\boldsymbol{m} = (m(\boldsymbol{x}_i)), i = 1, \cdots, N$, is a known N-dimensional column vector of Non-GP surrogate among training data; $\boldsymbol{C}$ is a known $N \times N$ correlation matrix among training data; $\boldsymbol{y}$ is a N-dimensional column observed vector among training data.

The estimation of σ^2 can be obtained by taking the partial derivative of Eq. (12) with respect to σ^2

$$\hat{\sigma}^2 = \frac{(\boldsymbol{y} - \boldsymbol{m})^T \boldsymbol{C}^{-1}(\boldsymbol{y} - \boldsymbol{m})}{N}. \tag{13}$$

Substituting Eq. (13) into Eq. (12), the maximum of log likelihood over $\hat{\sigma}^2$ is

$$-\frac{Nlog2\pi\hat{\sigma}^2 + log(det(\boldsymbol{C})) + N}{2}, \tag{14}$$

since Eq. (14) depends only on parameters within $\boldsymbol{C}$, thus above the maximum of log likelihood can be

$$-Nlog2\pi\hat{\sigma}^2 - log(det(\boldsymbol{C})), \tag{15}$$

3.1.2 Prediction Distribution

When all unknown hyperparameters are determined, then the prediction distribution at any untested point $\boldsymbol{x}$ can be obtained by using conditional distribution. The uncertainty (conditional variance) of prediction of Non-GP surrogates is

$$\hat{s}(\boldsymbol{x}) = \hat{\sigma}^2[1 - \boldsymbol{r}^T \boldsymbol{C}^{-1} \boldsymbol{r}], \tag{16}$$

where $\boldsymbol{r}$ is a known $N \times 1$ correlation matrix of the untested point $\boldsymbol{x}$ with training data.

3.2 Instantiation of UMP Framework

In UMP, any Non-GP surrogate can be considered to be the first term of the UMP. In this article, the first term will be instantiated with RBFN, QP, and SVM as a surrogate, respectively.

Algorithm 1 The pseudocode of the workflow of UMP

Input: Generate initial samples by LHS and evaluate by using exact functions, archive into an initial database.

Output: Output the best solution in the database.

while The computational budget is not exhausted **do**

Select τ samples in the database to train the Non-GP surrogate;

Evolve a population of NP individuals for T generations with a DE, evaluate by the Non-GP surrogate with **uncertainty measure method** (e.g., the proposed UMP in Eq. (16));

Select a candidate solution in the population by LCB to evaluate using exact functions;

Add the solution to the database, and update the database;

end

return Output the best solution in the database.

3.2.1 UMP with RBFN

The form of RBFN is described in Eq. (2). Here, the cubic kernel function is used as its activation function

$$\psi(\|\boldsymbol{x}-\boldsymbol{c}\|_p) = \|\boldsymbol{x}-\boldsymbol{c}\|_p^3, \tag{17}$$

where $\boldsymbol{c}$ is a center point vector, $p = 2$ in this article.

In this article, $2d+1$ cubic kernel functions are considered, based on the suggestion in Sprecher (1993). Based on the self-organizing method, $2d+1$ center point vectors are obtained by k-means algorithm (MacQueen, 1967).

3.2.2 UMP with Quadratic Polynomial

The quadratic polynomial (QP, second-order polynomial) is one of the most widely used polynomial regression models. Due to its simplicity and flexibility, QP is usually used as a surrogate and has a wide range of applications in various fields of science and engineering. It can be expressed as follows

$$\hat{f}(\boldsymbol{x}) = \beta_0 + \Sigma_{i=1}^{d}\beta_i x_i + \Sigma_{i=1,j=1,i\le j}^{d}\beta_{i,j}x_i x_j, \tag{18}$$

where β_0, β_i and $\beta_{i,j}$ are the coefficients to be estimated; $\boldsymbol{x} = [x_1, \ldots, x_d]^T$; QP is implemented using the Python tool-box (Pedregosa et al., 2011) in this article.

3.2.3 UMP with SVM

SVM is one of the regression techniques that have been introduced in Section II. In this article, the radial basis kernel function is adopted in SVM

$$\kappa(\boldsymbol{x},\boldsymbol{x}') =< \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') >= exp(-\gamma\|\boldsymbol{x}-\boldsymbol{x}'\|^2), \tag{19}$$

where the SVM is carried out using the Python tool-box (Pedregosa et al., 2011), the parameters γ is set as $'scale'$ in this article.

3.3 Workflow of UMP

The pseudocode for the workflow of UMP is presented in the Algorithm 1. Initially, $11d-1$ samples in the search space are generated using Latin hypercube sampling

(LHS) (Stein, 1987) and evaluated by exact functions. Then these samples are archived in an initial database. τ latest samples in the database are selected as training data to train the Non-GP surrogate. The Non-GP surrogate replace the exact functions in evolving a population of NP individuals for T generations with a DE. Then a potential candidate solution is selected in the population by using LCB and evaluated by exact functions. After that, the solution is added to the database. Finally, when the computational budget is exhausted, the best solution in the database is chosen as the output.

4 Results and Discussion

To investigate the performance of the proposed UMP, a set of experiments is carried out in both single and ensemble surrogates by Algorithm 1, respectively. The Non-GP surrogates involving RBFN, QP, and SVM are considered in this article.

For the single surrogate, two experiments are carried out. First, the experiment compares Non-GP surrogates with and without UMP, and they are named UMP/RBFN, UMP/QP, UMP/SVM, RBFN, QP, and SVM, respectively. Second, the UMP compares with the existing uncertainty method in Branke and Schmidt (2005), which is the distance from the solutions to the existing training data (DUM). The three algorithms with the UMP are named UMP/RBFN, UMP/QP, and UMP/SVM, and the compared algorithms are named DUM/RBFN, DUM/QP, and DUM/SVM.

Regarding ensemble surrogates, the proposed UMP compares with method U_{ens} which is the maximum difference between the outputs of the ensemble members (Wang et al., 2017) and VUM which is variance of predictions output by the base surrogates of the ensemble (Guo et al., 2018), respectively. In this article, the ensemble surrogates consist of three surrogates: RBFN, QP, and SVM. These algorithms with the proposed method and two compared methods are named UMP/ensemble, U_{ens}/ensemble, and VUM/ensemble, respectively.

4.1 Parameter Settings

There are several parameters in experiments. The setting of these parameters is given below.

(1) The computational budget with exact function evaluations $FEs = 100$ was performed in this article, based on the assumption that the optimization algorithm is only allowed to evaluate a small number of candidate solutions during optimization. The number of the run was 25.

(2) DE parameters: DE/rand/1/bin was employed in this article. The evolution generations $T = 100$, population size $NP = 20$, scaling factor $F = 0.5$, and the crossover rate $CR = 0.9$.

(3) Initial samples $11d - 1$ were randomly generated by LHS.

(4) The range of values for parameters $\boldsymbol{\theta}$ was $[1.0e - 6, 20]$.

(5) Training data $\tau = 50$ for dimension $d = 2$, $\tau = 11d - 1$ for $d = 5, 10$ was considered. τ training data in the database was selected under considering both the quality and the computational cost of Non-GP surrogate.

4.2 Test Problems

The effectiveness of the proposed method is verified on benchmark problems CEC 2014 (Liu et al., 2014) with 2, 5, and 10 dimensions. The benchmark problems are listed in Table 1.

Table 1: Test problems.

Problem	Objective function name	f^*	Property
$F1$	Shifted Sphere	0	Unimodal
$F2$	Shifted Ellipsoid	0	Unimodal
$F3$	Shifted and Rotated Ellipsoid	0	Unimodal
$F4$	Shifted Step	0	Unimodal, Discontinuous
$F5$	Shifted Ackley	0	Multi-modal
$F6$	Shifted Griewank	0	Multi-modal
$F7$	Shifted and Rotated Rosenbrock	0	Multi-modal with very narrow valley
$F8$	Shifted and Rotated Rastrigin	0	Very complicated multi-modal

4.3 Comparison on Single Non-GP Surrogate

4.3.1 Effect of the UMP

To investigate the effectiveness of the proposed UMP, a set of comparative experiments are carried out for algorithms with and without UMP. Table 2 presents all algorithms' average best fitness values on test problems with 2, 5, and 10 dimensions. Figures 1, 2, and 3 present the comparison of convergence curves of different algorithms on $F1$, $F5$, and $F8$ test problems with different dimensions, respectively.

From Table 2 and Figures 1–3, the results of UMP/RBFN, UMP/QP, and UMP/SVM are significantly better than those of the other algorithms in most test problems. The performance of the three algorithms (UMP/RBFN, UMP/QP, and UMP/SVM) is always better than the other on both unimodal or multimodal problems, especially in 5- and 10-dimensional test problems. The results are mainly attributed to the fact that there is uncertainty information in the three algorithms that can be used to guide the search of algorithms and the update of surrogates. The test problems are more complex as the number of dimensions increases, and there are many local optima for multimodal problems. The uncertainty information is able to strengthen the exploration ability of algorithms and improve the accuracy of the surrogates.

4.3.2 Comparison with Peer Algorithms

To further investigate the effectiveness of the proposed UMP, the proposed UMP is compared with the uncertainty method DUM, which is shown in Eq. (20). Table 3 presents the average best fitness values obtained by proposed algorithms UMP/RBFN, UMP/QP, and UMP/SVM, and the compared algorithms DUM/RBFN, DUM/QP, and DUM/SVM. Figures 4, 5, and 6 present the comparison of convergence curves of different algorithms on $F1$, $F5$, and $F8$ with different dimensions, respectively.

$$U(\boldsymbol{x}) = \frac{1}{\sum_{i=1}^{L} \frac{1}{d_{\boldsymbol{x}\boldsymbol{x}_i'}}}, \tag{20}$$

where $U(\boldsymbol{x})$ represents the uncertainty of prediction of a solution $\boldsymbol{x}$, $d_{\boldsymbol{x}\boldsymbol{x}_i'}$ is the Euclidean distance from solution $\boldsymbol{x}$ to solution $\boldsymbol{x}_i'$ in the training data τ, and L is the number of solutions in the neighborhood used for estimation; L is equal to the number of training data τ in this article.

The Table 3 and Figures 4–6 show that UMP/RBFN, UMP/QP, and UMP/SVM achieve significantly better performance than the other algorithms on most test problems. There is a significant difference in 10 dimensional test problems, especially for $F3$,

Table 2: Comparing the averages fitness values (shown as Avg ± Std) of UMP/RBFN, UMP/QP, UMP/SVM, RBFN, QP, and SVM. Statistically significant results evaluated using a Friedman test.

Problem	d	UMP/RBFN	RBFN	UMP/QP	QP	UMP/SVM	SVM
F1	2	**9.8806E-3** ± 0.0156	0.0642 ± 0.1018	**1.4462E-4** ± 2.4305E-4	2.8772 ± 2.8031	**6.0986E-4** ± 7.6748E-4	0.8707 ± 1.5511
	5	**0.0331** ± 0.0328	0.2288 ± 0.1966	**0.7640** ± 0.7112	54.0371 ± 23.7337	**0.0264** ± 0.0169	25.1719 ± 16.9316
	10	**1.9803** ± 1.0066	9.6927 ± 5.0127	**67.9611** ± 21.5325	287.0700 ± 76.2774	**21.5962** ± 11.4967	87.2006 ± 37.7374
F2	2	**0.0182** ± 0.0342	0.2342 ± 0.5029	**3.0383E-4** ± 2.5810E-4	2.8759 ± 2.6865	**1.0445E-3** ± 1.3433E-3	5.2747 ± 8.8904
	5	**0.4057** ± 0.3093	2.3933 ± 1.7200	**14.0614** ± 19.1254	155.4160 ± 68.8183	**0.0799** ± 0.0468	173.4310 ± 82.1052
	10	**32.4434** ± 16.5484	129.7015 ± 51.3120	**307.0399** ± 97.6482	1425.5204 ± 393.2456	**109.8823** ± 53.2732	555.1530 ± 268.1075
F3	2	**0.0437** ± 0.0915	1.2991 ± 1.8210	**9.9812E-3** ± 0.0320	1.3025 ± 1.2417	**0.0011** ± 0.0010	5.4055 ± 7.0946
	5	**0.4611** ± 0.4911	3.2056 ± 4.6624	**5.6759** ± 5.0457	39.0094 ± 21.5417	**0.0307** ± 0.0185	38.9898 ± 25.2550
	10	**243.0518** ± 84.7628	823.7360 ± 378.8218	**477.9377** ± 182.8839	1921.1620 ± 503.9823	**559.6709** ± 316.1527	2451.7201 ± 795.8515
F4	2	**0.0** ± 0.0	**0.0** ± 0.0	**0.0** ± 0.0	2.2400 ± 2.1029	**0.0** ± 0.0	0.6800 ± 0.6144
	5	**0.1200** ± 0.3249	0.2000 ± 0.4000	**4.4000** ± 4.9799	49.0000 ± 24.5715	**0.7600** ± 1.0688	26.2000 ± 24.6398
	10	**2.5200** ± 1.0998	9.5200 ± 4.3185	**65.2000** ± 22.7385	230.8000 ± 77.1305	**23.5200** ± 10.9986	76.1600 ± 34.7754
F5	2	**2.7644** ± 1.0851	10.4969 ± 4.7438	**2.7476** ± 0.8620	8.0343 ± 2.2641	**1.8426** ± 0.6829	2.2079 ± 1.1024
	5	**3.6208** ± 0.7392	12.9774 ± 1.5847	**4.0928** ± 0.912	16.1825 ± 1.9481	**3.5112** ± 0.5640	6.3787 ± 2.4737
	10	**4.2216** ± 0.3694	13.2189 ± 1.1633	**10.6836** ± 6.2570	18.4682 ± 1.0169	**3.7534** ± 0.3806	9.8341 ± 1.4496
F6	2	**0.1410** ± 0.0035	0.1600 ± 0.3545	**0.4851** ± 0.2165	1.1618 ± 0.7925	**0.0602** ± 0.0401	0.1971 ± 0.1836
	5	**0.6092** ± 0.2302	0.7121 ± 0.8439	**1.1118** ± 0.2986	16.7889 ± 7.6244	**0.5397** ± 0.1432	3.9278 ± 3.3254
	10	**1.1185** ± 0.0837	1.2103 ± 0.1364	**14.2539** ± 4.4617	75.7570 ± 21.1887	**3.7237** ± 1.0159	18.4265 ± 6.9328
F7	2	**0.1219** ± 0.1055	0.2861 ± 0.2670	**0.0732** ± 0.0671	0.0773 ± 0.0712	**0.0131** ± 0.0142	0.2308 ± 0.2212
	5	**43.3089** ± 30.4430	70.2126 ± 47.2680	**66.0930** ± 34.6149	72.0384 ± 30.1393	**67.1543** ± 27.5345	126.1766 ± 79.2998
	10	**204.2603** ± 65.2703	316.0039 ± 125.3541	**353.6906** ± 159.2818	728.8562 ± 328.2364	**708.1847** ± 356.1181	719.1093 ± 365.0644
F8	2	**3.1274** ± 1.4961	3.3412 ± 3.0295	**3.7549** ± 2.0459	3.8671 ± 1.3504	**2.3542** ± 1.9835	3.1703 ± 2.2549
	5	**18.5786** ± 3.9201	19.3769 ± 4.4019	**22.5314** ± 7.7902	29.2176 ± 5.6230	**21.0797** ± 4.3804	22.6443 ± 7.0308
	10	**59.4989** ± 11.4535	59.9236 ± 6.4440	**76.0060** ± 16.9585	96.6026 ± 16.0349	**72.4144** ± 13.7050	90.2328 ± 12.0011
Average rank		**1.98**	3.35	**3.35**	5.46	**2.02**	4.83

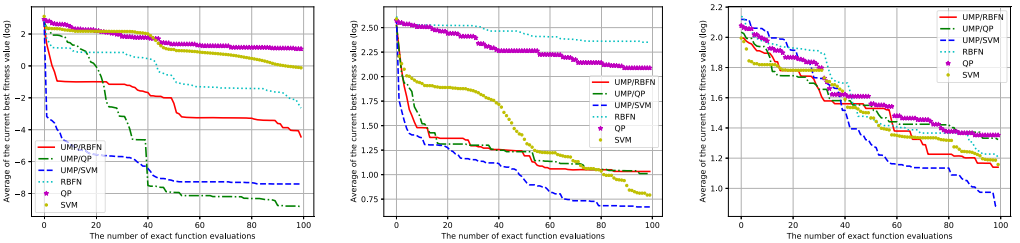


Figure 1: Comparison of with and without UMP on convergence curves for $F1$, $F5$, and $F8$ on 2d, respectively.

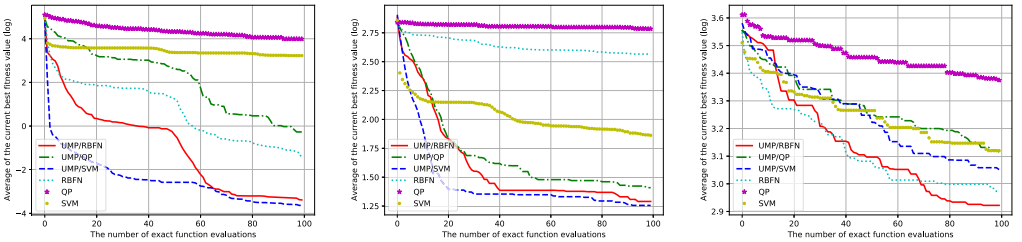


Figure 2: Comparison of with and without UMP on convergence curves for $F1$, $F5$, and $F8$ on 5d, respectively.

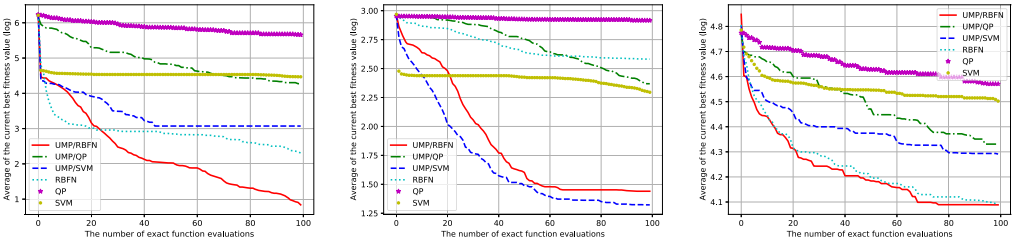


Figure 3: Comparison of with and without UMP on convergence curves for $F1$, $F5$, and $F8$ on 10d, respectively.

$F5$, and $F7$, which maybe due to the accuracy of DUM that is unreliable. Thus, the DUM cannot efficiently guide the search of algorithms when there are many local optima on $F5$ and $F7$ in 10-dimensional test problems, and it cannot accurately fit the fitness landscape of $F3$, $F5$, and $F7$ in 10-dimensional test problems. Instead, the proposed UMP still has good performance in these problems with 10 dimension.

A comparison experiment is carried out to analyze further the proposed method on $F8$, with the results shown in Figure 7. In the figure, the RBFN surrogate is used as a regression function to approximate the exact function, and the UMP and DUM methods are used to estimate the uncertainty of prediction of RBFN. From Figure 7a, there is a significant error between the prediction values of RBFN and the exact function. The proposed UMP is able to measure the error more accurately than DUM. Besides, the values adjusting to the prediction of RBFN from the proposed UMP are better at

Table 3: Comparing the averages fitness values (shown as Avg ± Std) of UMP/RBFN, UMP/QP, UMP/SVM, DUM/RBFN, DUM/QP, and DUM/SVM. Statistically significant results evaluated using a Friedman test.

Problem	d	UMP/RBFN	DUM/RBFN	UMP/QP	DUM/QP	UMP/SVM	DUM/SVM
F1	2	**9.8806E-3** ± 0.0156	0.0502 ± 0.0802	**1.4462E-4** ± 2.4305E-4	4.1442 ± 3.3661	**6.0986E-4** ± 7.6748E-4	0.0180 ± 0.0232
	5	**0.0331** ± 0.0328	0.1686 ± 0.1147	**0.7640** ± 0.7112	140.5686 ± 65.2623	**0.0264** ± 0.0169	26.6025 ± 25.5855
	10	**1.9803** ± 1.0066	9.4756 ± 4.5445	**67.9611** ± 21.5325	477.5787 ± 130.4803	**21.5962** ± 11.4967	84.9443 ± 29.7331
F2	2	**0.0182** ± 0.0342	0.2449 ± 0.3725	**3.0383E-4** ± 2.5810E-4	6.7903 ± 6.7277	**1.1499E-3** ± 1.0445E-3	0.3620 ± 0.7392
	5	**0.4057** ± 0.3093	4.1801 ± 3.7846	**14.0614** ± 19.1254	365.6379 ± 146.3157	**0.0799** ± 0.0468	148.4953 ± 124.1047
	10	**32.4434** ± 16.5484	119.4572 ± 49.2795	**307.0399** ± 97.6482	2484.1205 ± 510.5520	**109.8823** ± 53.2732	638.3514 ± 180.0408
F3	2	**0.0437** ± 0.0915	0.5505 ± 0.6879	**9.9812E-3** ± 0.0320	4.7287 ± 2.9526	**0.0053** ± 0.0069	0.2930 ± 0.5269
	5	**0.4611** ± 0.4911	2.0510 ± 1.1517	**5.6759** ± 5.0457	110.6858 ± 48.6745	**0.0307** ± 0.0185	35.7253 ± 33.3522
	10	**243.0518** ± 84.7628	848.8346 ± 268.9680	**477.9377** ± 182.8839	3141.0693 ± 956.7205	**559.6709** ± 316.1527	2204.2445 ± 1091.0657
F4	2	**0.0** ± 0.0	**0.0** ± 0.0	**0.0** ± 0.0	4.0800 ± 3.6212	**0.0** ± 0.0	**0.0** ± 0.0
	5	**0.1200** ± 0.3249	0.1600 ± 0.3666	**4.4000** ± 4.9799	141.1600 ± 55.9783	**0.7600** ± 1.0688	31.6800 ± 26.8264
	10	**2.5200** ± 1.0998	8.6000 ± 3.1874	**65.2000** ± 22.7385	439.3200 ± 112.3308	**23.5200** ± 10.9986	76.6000 ± 30.6111
F5	2	**2.7644** ± 1.0851	9.7490 ± 3.8671	**2.7476** ± 0.8620	8.1565 ± 2.7408	1.8426 ± 0.6829	**0.9289** ± 0.5018
	5	**3.6208** ± 0.7392	11.8476 ± 2.1805	**4.0928** ± 0.9124	17.0522 ± 1.5073	**3.5112** ± 0.5640	6.0516 ± 1.4182
	10	**4.2216** ± 0.3694	12.7883 ± 1.5767	**10.6836** ± 6.2570	19.2076 ± 0.7207	**3.7534** ± 0.3806	9.1504 ± 1.2157
F6	2	**0.1410** ± 0.0035	0.1494 ± 0.0676	**0.4851** ± 0.216	2.0738 ± 1.0517	**0.0602** ± 0.0401	0.1313 ± 0.1128
	5	**0.6092** ± 0.2302	0.6143 ± 0.1708	**1.1118** ± 0.2986	33.2851 ± 11.7161	**0.5397** ± 0.1432	5.2599 ± 4.2937
	10	**1.1185** ± 0.0837	1.1323 ± 0.0466	**14.2539** ± 4.4617	117.7120 ± 20.8241	**3.7237** ± 1.0159	17.1081 ± 6.6650
F7	2	**0.1219** ± 0.1055	0.2439 ± 0.2661	**0.0732** ± 0.0671	0.1221 ± 0.1160	**0.0131** ± 0.0142	0.0974 ± 0.0516
	5	**43.3089** ± 30.4430	88.9265 ± 47.3513	**66.0930** ± 34.6149	113.1713 ± 55.7217	**67.1543** ± 27.5345	104.1294 ± 65.0487
	10	**204.2603** ± 65.2703	311.2954 ± 159.9181	**353.6906** ± 159.2818	1876.2670 ± 884.2830	**708.1847** ± 356.1181	719.7931 ± 303.3686
F8	2	**3.1274** ± 1.4961	3.3755 ± 2.4286	**3.7549** ± 2.0459	4.1802 ± 2.0134	**2.3542** ± 1.9835	2.4814 ± 2.0600
	5	**18.5786** ± 3.9201	18.6270 ± 5.6959	**22.5314** ± 7.7902	31.8924 ± 8.3335	**21.0797** ± 4.3804	21.1876 ± 6.2968
	10	**59.4989** ± 11.4535	60.8467 ± 6.7949	**76.0060** ± 16.9585	123.9444 ± 13.2013	**72.4144** ± 13.7050	83.7737 ± 13.1376
Average rank		**2.08**	3.42	**3.38**	5.92	**2.04**	4.17

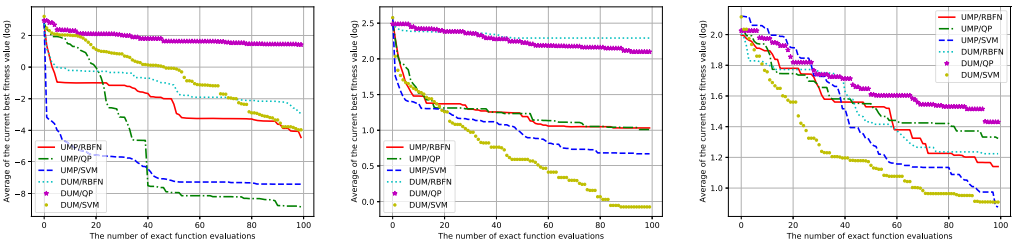


Figure 4: Comparison of UMP and DUM on convergence curves for $F1$, $F5$, and $F8$ on 2d, respectively.

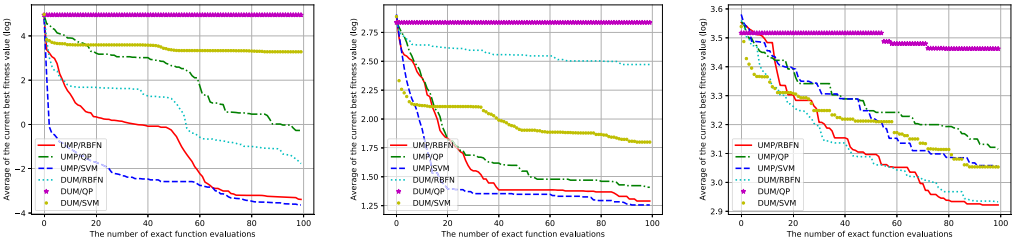


Figure 5: Comparison of UMP and DUM on convergence curves for $F1$, $F5$, and $F8$ on 5d, respectively.

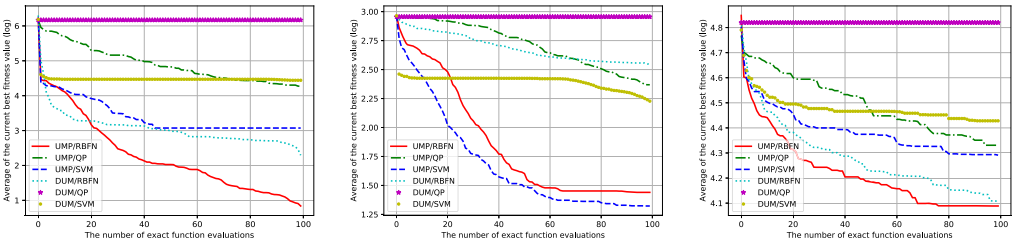
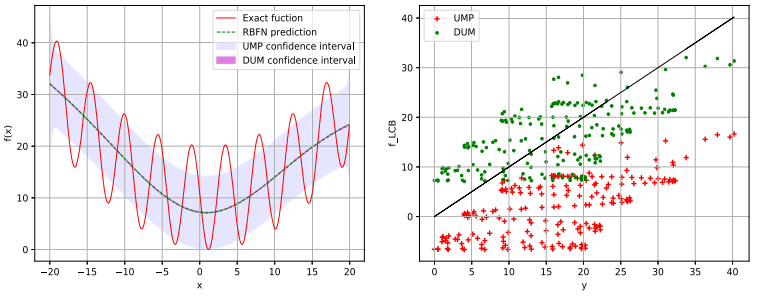


Figure 6: Comparison of UMP and DUM on convergence curves for $F1$, $F5$, and $F8$ on 10d, respectively.

approximating the exact function values than DUM, according to the assumption of random field model as Eq. (10). Therefore, the proposed UMP has better performance in the measurement of uncertainty for the prediction of RBFN than DUM. The validity of the proposed UMP also is investigated in Figure 7b, and it shows that f_{LCB} of UMP is a better approximation to the lower bound for the exact function values than that of DUM. The DUM method indicates the crowded degree of the neighborhood of a solution. Thus it cannot accurately measure the uncertainty of prediction of RBFN.

4.4 Comparison on Ensemble Surrogates

An experiment is carried out to show the effectiveness of the proposed UMP on ensemble surrogates case. The proposed method is compared with the uncertainty method U_{ens} in Eq. (7) and method VUM in Eq. (8), respectively. Table 4 presents the average



(a) (b)

Figure 7: (a) Illustration of UMP and DUM methods on a 1-d toy example with $F8$ function; the number of training points 100 is considered; red curve represents the exact function; green dashed curve represents the prediction from RBFN; shaded regions represent the confidence interval of the prediction of RBFN; (b) $y - f_{LCB}$ plot on $F8$ in single surrogate RBFN case.

best fitness values obtained by UMP/ensemble, U_{ens}/ensemble, and VUM/ensemble algorithms on a set of test problems. Figures 8, 9, and 10 present the comparison of convergence curves of different algorithms on $F1$, $F5$, and $F8$ with different dimensions, respectively.

From Table 4 and Figures 8–10, the UMP/ensemble significantly outperforms U_{ens}/ensemble and VUM/ensemble on most test problems. In the results, the performance of VUM/ensemble is worst. The performance of UMP/ensemble and U_{ens}/ensemble is similar on $F4$ in 2 and 5 dimensions, and U_{ens}/ensemble is slightly better than UMP/ensemble on $F6$, $F7$, and $F8$ in 5 dimension. However, UMP/ensemble is worse than U_{ens}/ensemble on $F8$ in 10 dimension, probably because the number of computational budget $FEs = 100$ is too small. Most of the reevaluation solutions are consumed due to a large degree of uncertainty, resulting in less opportunity for UMP/ensemble to exploit the search space sufficiently.

The effectiveness is further analyzed on ensemble surrogates case in Figure 11. In the figure, the ensemble of RBFN, QP, and SVM surrogates is used as regression function to approximate the exact $F8$ function, and the UMP, VUM, and U_{ens} methods are used to estimate the uncertainty of prediction of the ensemble, respectively. From Figure 11a, there is a significant error between the prediction values of the ensemble and the exact function. The proposed UMP is best to approximate the error, and the uncertainty of the ensemble from VUM and U_{ens} methods is similar in general. Besides, the values obtained by UMP are the best approximation of the exact function values than VUM and U_{ens} in general, according to the assumption of random field model as Eq. (10). Figure 11b also verifies f_{LCB} of UMP is a best approximation to the lower bound for the exact function values.

Regarding the uncertainty method U_{ens}, it is maximum difference among the outputs of the ensemble members for the prediction of a solution. In essence, this method describes the prediction difference among ensemble members in a solution. Likewise, for the method VUM, the uncertainty of prediction of a solution is measured by the variance of the prediction output by the base surrogates of the ensemble. This method indicates the average squared deviation of the base surrogates about the output of the ensemble surrogates. From the probability and statistic viewpoint, these methods

Table 4: Comparing the averages fitness values (shown as Avg ± Std) of UMP/ensemble, U_{ens}/ensemble, and VUM/ensemble. Statistically significant results evaluated using a Friedman test.

Problem	d	UMP/ensemble	VUM/ensemble	U_{ens}/ensemble
F1	2	**7.7161E-5** ± 1.5035E-4	1.3604 ± 1.2453	9.3382E-3 ± 0.0197
	5	**0.0011** ± 0.0012	12.1513 ± 8.6543	0.1555 ± 0.0832
	10	**1.0012** ± 0.7830	23.5616 ± 9.8029	9.3609 ± 3.5025
F2	2	**2.0820E-4** ± 2.2511E-4	3.7257 ± 2.9982	0.0810 ± 0.1031
	5	**0.0526** ± 0.0614	77.1360 ± 38.0107	2.6106 ± 2.0893
	10	**33.0773** ± 25.3703	284.9453 ± 95.0346	141.5993 ± 40.6508
F3	2	**4.1281E-4** ± 5.1014E-4	3.4762 ± 2.0919	0.6851 ± 1.3881
	5	**0.1005** ± 0.1455	28.3199 ± 11.8856	2.5193 ± 1.5458
	10	**537.8770** ± 150.8156	1204.2153 ± 381.1870	889.4860 ± 271.9350
F4	2	**0.0** ± 0.0	1.3200 ± 1.3181	**0.0** ± 0.0
	5	**0.1200** ± 0.3249	12.600 ± 11.0381	**0.1200** ± 0.3249
	10	**3.2400** ± 1.9448	21.0000 ± 8.4332	10.2000 ± 4.8249
F5	2	**2.7002** ± 1.1667	10.3965 ± 4.9964	3.8192 ± 1.8471
	5	**3.5092** ± 0.7333	13.8540 ± 2.8998	8.4735 ± 1.5095
	10	**5.5079** ± 0.5751	13.1730 ± 0.9270	10.8165 ± 1.1127
F6	2	**0.0872** ± 0.0544	0.6791 ± 0.3800	0.1351 ± 0.0789
	5	0.6518 ± 0.2003	1.3107 ± 0.3293	**0.6159** ± 0.1419
	10	**1.0563** ± 0.1378	1.3724 ± 0.3217	1.4022 ± 0.3057
F7	2	**0.0076** ± 0.0090	0.1469 ± 0.1503	0.2916 ± 0.2351
	5	63.8693 ± 30.8029	81.6944 ± 35.7565	**59.3588** ± 38.5356
	10	**235.0830** ± 119.8799	585.8805 ± 231.1875	334.1423 ± 157.4397
F8	2	**1.7132** ± 1.1547	3.3034 ± 1.8626	2.7242 ± 1.3798
	5	19.9698 ± 5.5287	24.4794 ± 5.4224	**17.1605** ± 4.8970
	10	69.4359 ± 7.8951	74.3970 ± 11.7897	**58.7404** ± 8.5256
Average rank		**1.27**	2.83	1.90

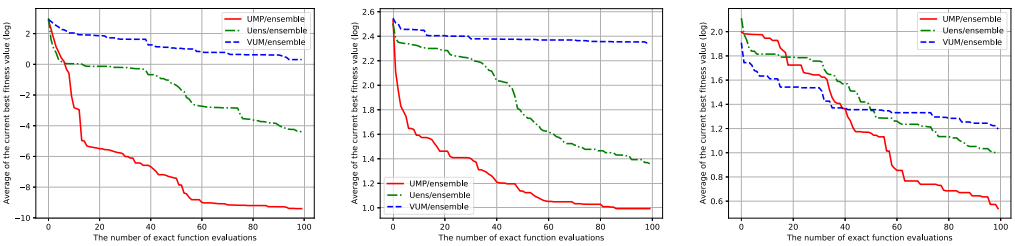


Figure 8: Comparison of UMP, U_{ens} and VUM on convergence curves for $F1$, $F5$, and $F8$ on 2d, respectively.

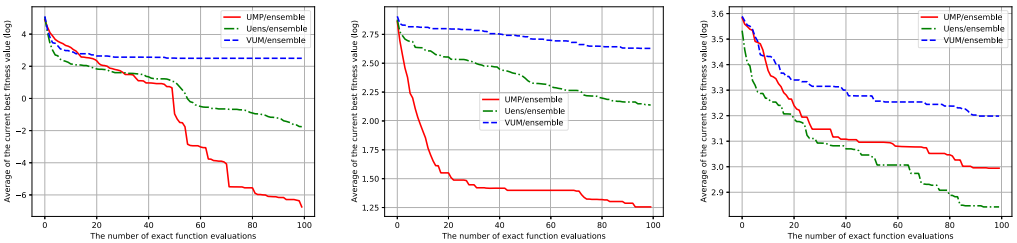


Figure 9: Comparison of UMP, U_{ens} and VUM on convergence curves for $F1$, $F5$, and $F8$ on 5d, respectively.

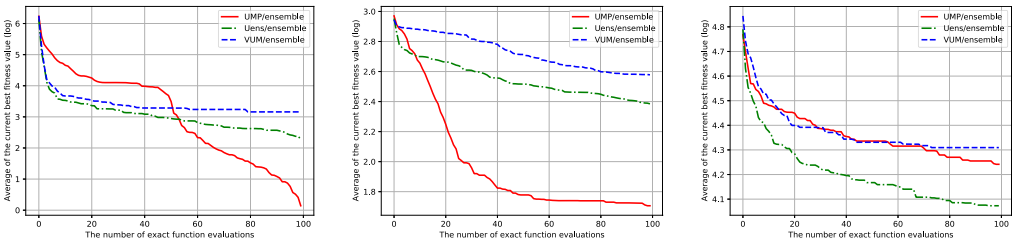
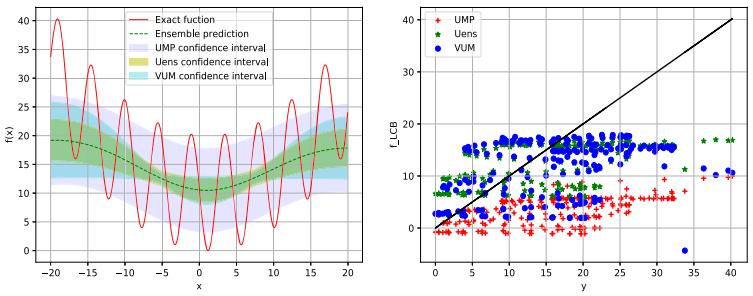


Figure 10: Comparison of UMP, U_{ens} and VUM on convergence curves for $F1$, $F5$, and $F8$ on 10d, respectively.



(a) (b)

Figure 11: (a) Illustration of UMP, U_{ens} and VUM methods on a 1-d toy example with $F8$ function; the number of training points 100 is considered; red curve represents the exact function; green dashed curve represents the ensemble prediction from the weight sum of prediction of RBFN, QP and SVM; shaded regions represent confidence interval of the ensemble prediction; (b) $y - f_{LCB}$ plot on $F8$ in ensemble surrogates case.

measuring the uncertainty of prediction can be insufficient. Therefore, these compared methods are potentially unreliable in measuring the uncertainty of the prediction of surrogates.

5 Conclusion

This article mainly addresses the issue that there is no theoretical method to measure the uncertainty of prediction of Non-GP surrogates. This article proposes a theoretical

method to measure the uncertainty. In this method, a stationary random field with a known zero mean is used to measure the uncertainty of prediction of Non-KGP surrogates. The method's effectiveness has been verified based on some experiments in single and ensemble surrogates cases. The experimental results demonstrate that the proposed method is more promising than other methods on a set of test problems.

Although the performance of the proposed method is competitive, the computational cost of this method is higher than others. The computational cost of UMP, DUM, U_{ens} and VUM is $\mathcal{O}(N^3)$, $\mathcal{O}(N)$, $\mathcal{O}(1)$, and $\mathcal{O}(1)$, respectively. N is the number of training samples. Thus, the main limitation of the method is that it needs more computational cost in measuring the uncertainty for the prediction of Non-GP surrogates. Therefore, our future work is to solve this drawback using transfer learning.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62076226, in part by the Fundamental Research Funds for the Central Universities China University of Geosciences (Wuhan) under Grant *CUGGC*02, and in part by the 111 project under Grant *B*17040.

References

Branke, J., and Schmidt, C. (2005). Faster convergence by means of fitness estimation. *Soft Computing*, 9(1):13–20. 10.1007/s00500-003-0329-4

Braun, J., Krettek, J., Hoffmann, F., and Bertram, T. (2009). Multi-objective optimization with controlled model assisted evolution strategies. *Evolutionary Computation*, 17(4):577–593. 10.1162/evco.2009.17.4.17408

Chugh, T., Jin, Y., Miettinen, K., Hakanen, J., and Sindhya, K. (2016). A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 22(1):129–142. 10.1109/TEVC.2016.2622301

Chugh, T., Sindhya, K., Hakanen, J., and Miettinen, K. (2019). A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Computing*, 23(9):3137–3166. 10.1007/s00500-017-2965-0

Coelho, R. F., and Bouillard, P. (2011). Multi-objective reliability-based optimization with stochastic metamodels. *Evolutionary Computation*, 19(4):525–560. 10.1162/EVCO_a_00034

Cristianini, N., and Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge: Cambridge University Press.

Emmerich, M., Giannakoglou, K., and Naujoks, B. (2006). Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439. 10.1109/TEVC.2005.859463

Emmerich, M., Giotis, A., Özdemir, M., Bäck, T., and Giannakoglou, K. (2002). Metamodel-assisted evolution strategies. In *2002 International Conference on Parallel Problem Solving from Nature*, pp. 361–370.

Gal, Y., and Ghahramani, Z. (2015). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *2015 International Conference on Machine Learning*, pp. 1050–1059.

Guo, D., Jin, Y., Ding, J., and Chai, T. (2018). Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems. *IEEE Transactions on Cybernetics*, 49(3):1012–1025. 10.1109/TCYB.2018.2794503

Hart, E., Ross, P., and Nelson, J. (1998). Solving a real-world problem using an evolving heuristically driven schedule builder. *Evolutionary Computation*, 6(1):61–80. 10.1162/evco.1998.6.1.61

Hutter, F., Kotthoff, L., and Vanschoren, J. (2019). *Automated machine learning: Methods, systems, challenges*. Berlin: Springer Nature.

Jin, Y. (2011). Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70. 10.1016/j.swevo.2011.05.001

Jin, Y., Olhofer, M., and Sendhoff, B. (2000). On evolutionary optimization with approximate fitness functions. In *2000 IEEE Genetic and Evolutionary Computation Conference (GECCO)*, pp. 786–793.

Jin, Y., Wang, H., Chugh, T., Guo, D., and Miettinen, K. (2018). Data-driven evolutionary optimization: An overview and case studies. *IEEE Transactions on Evolutionary Computation*, 23(3):442–458. 10.1109/TEVC.2018.2869001

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492. 10.1023/A:1008306431147

Lewis, R. M., Torczon, V., and Trosset, M. W. (2000). Direct search methods: Then and now. *Journal of Computational and Applied Mathematics*, 124(1-2):191–207. 10.1016/S0377-0427(00)00423-4

Li, C., Yang, S., and Yang, M. (2014). An adaptive multi-swarm optimizer for dynamic optimization problems. *Evolutionary Computation*, 22(4):559–594. 10.1162/EVCO_a_00117

Liu, B., Chen, Q., Zhang, Q., Liang, J., Suganthan, P. N., and Qu, B. (2014). Problem definitions and evaluation criteria for computational expensive optimization. In *2014 IEEE Congress on Evolutionary Computation*, pp. 2081–2088.

Liu, Y., Yao, X., and Higuchi, T. (2000). Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation*, 4(4):380–387. 10.1109/4235.887237

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *1967 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297.

Michalewicz, Z., and Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4(1):1–32. 10.1162/evco.1996.4.1.1

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., and Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Preen, R. J., and Bull, L. (2016). Design mining interacting wind turbines. *Evolutionary Computation*, 24(1):89–111. 10.1162/EVCO_a_00144

Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K. (2005). Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1–28. 10.1016/j.paerosci.2005.02.001

Sprecher, D. A. (1993). A universal mapping for Kolmogorov's superposition theorem. *Neural Networks*, 6(8):1089–1094. 10.1016/S0893-6080(09)80020-8

Stein, M. (1987). Large sample properties of simulations using Latin hypercube sampling. *Technometrics*, 29(2):143–151. 10.1080/00401706.1987.10488205

Tong, H., Huang, C., Liu, J., and Yao, X. (2019). Voronoi-based efficient surrogate-assisted evolutionary algorithm for very expensive problems. In *2019 IEEE Congress on Evolutionary Computation*, pp. 1996–2003.

Torczon, V., and Trosset, M. W. (1998). Using approximations to accelerate engineering design optimization. In *1998 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, p. 4800.

Ulmer, H., Streichert, F., and Zell, A. (2003). Evolution strategies assisted by Gaussian processes with improved preselection criterion. In *2003 IEEE Congress on Evolutionary Computation*, pp. 692–699.

Wang, H., Jin, Y., and Doherty, J. (2017). Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. *IEEE Transactions on Cybernetics*, 47(9):2664–2677. 10.1109/TCYB.2017.2710978

Wang, H., Jin, Y., and Jansen, J. O. (2016). Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system. *IEEE Transactions on Evolutionary Computation*, 20(6):939–952. 10.1109/TEVC.2016.2555315

Wang, W., Liu, H.-L., and Tan, K. C. (2022). A surrogate-assisted differential evolution algorithm for high-dimensional expensive optimization problems. *IEEE Transactions on Cybernetics*, 1–13. 10.1109/TCYB.2022.3175533

Zhan, D., and Xing, H. (2021). A fast Kriging-assisted evolutionary algorithm based on incremental learning. *IEEE Transactions on Evolutionary Computation*, 25(5):941–955. 10.1109/TEVC.2021.3067015

Zhang, F., Mei, Y., Nguyen, S., Zhang, M., and Tan, K. C. (2021). Surrogate-assisted evolutionary multitask genetic programming for dynamic flexible job shop scheduling. *IEEE Transactions on Evolutionary Computation*, 25(4):651–665. 10.1109/TEVC.2021.3065707

Zhang, M., Li, H., Pan, S., Lyu, J., Ling, S., and Su, S. (2021). Convolutional neural networks based lung nodule classification: A surrogate-assisted evolutionary algorithm for hyperparameter optimization. *IEEE Transactions on Evolutionary Computation*, 25(5):869–882. 10.1109/TEVC.2021.3060833